iris UNIVERSE

Silicon Graphics

summer 1988

the IRIS community magazine

ADD NEW EXPRESSION TO YOUR CREATIVE POTENTIAL !

RAN/Pierre-Jean Liéveau

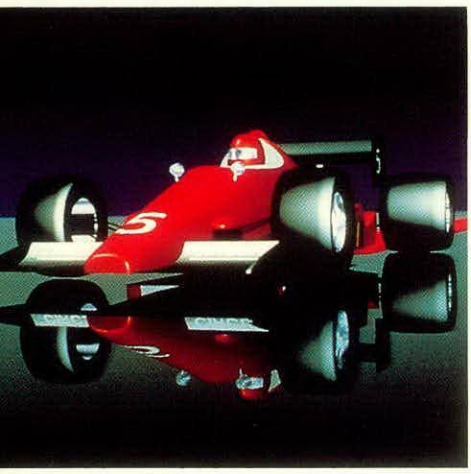
EXPLORE

THE LEADING EUROPEAN TURNKEY 3D ANIMATION SOFTWARE

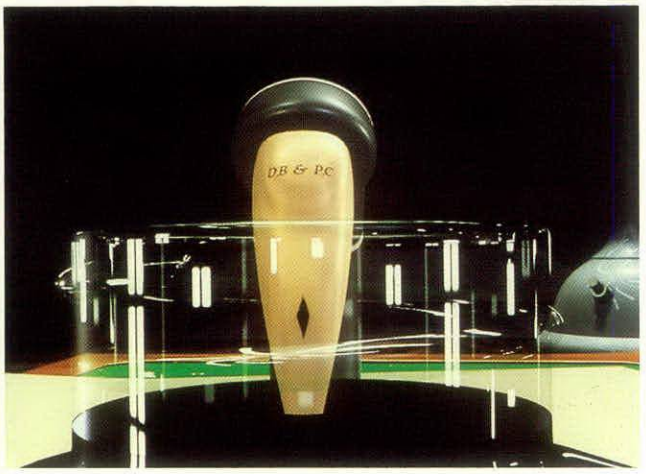
EXPLORE offers you exciting new scope in creating, manipulating and animating your computer drawn images. A whole realm of possibilities for rapid image movement, giving true-to-life expression to 3D animations. Best of all, EXPLORE's easy user/system dialogue allows you to work naturally. It's totally mouse and menu driven, giving you full visual control at all times.

EXPLORE is available for the IRIS SiliconGraphics™ range, including the new 4D/70, and in a PC version - thanks to a Logigraphics™ card -. EXPLORE can be interfaced with most CAD programmes and all visual supports : analog and digital screens, 35 mm film, high definition hard copy for printing...

Because your imagination is boundless, you need a complete computer graphics system. Discover EXPLORE and add new expression to your creative potential.



La Cinq/TDI



"Le Stylo" (The Pen) Pixel INA Award/Imagina 88
Eurocitel/Daniel Borenstein, Paul Coudsi

Distributors

HOLLAND - DENMARK - BELGIUM		
HIGH SKY B.V.	AMSTERDAM	Tel : 20 272771
ITALY		
SIGECO	BRESCIA	Tel : 030 51 349
AUTOMA	GENOVA	Tel : 010 28 10 94 / 010 29 17 86
SPAIN		
TELSON	MADRID	Tel : (34) 1 413 44 63 + 1 413 66 62
U.S.A. - CANADA		
RAINBOW IMAGES INC.	SAN JOSE, CA	Tel : (408) 971-7600

TDI SYSTÈMES

22 RUE HEGESIPPE- MOREAU
75018 PARIS FRANCE
TELEPHONE (33) 1/43 87 58 58
FAX (33) 1/ 43 87 61 11
TELEX TCSF 204 780 ADIG

qd EQUATION

it's a colorful UNIVERSE

With this SIGGRAPH edition, the IRIS Universe discovers the beauty of full color. Now we are able to publish the many high quality digital images you produce and submit. It's a pleasure to look at them on the screen, and it's only one more step to convert your RGB data directly into the print realm of cyan, magenta, yellow, and black (YMCK) by digital color separation. None of this would have been possible without the support of our Geometry Partners. Their advertisements make the IRIS Universe even more informative. We'd like to thank all of you.
Many of us will come together at the International IRIS User Forum to be held at SIGGRAPH in Atlanta, Georgia. We hope to see you there. Keep the colorful spirit alive!

International IRIS User Forum
Marriott Hotel, Imperial Ballroom,
Atlanta, Georgia
Wednesday, August 3, 1988
11 am - 7:30 pm

cover pic

the garden of L
©1988 by Przemyslaw Prusinkiewicz, Dave Fracchia, Jim Hanan and Debbie Fowler
Dept. of Computer Science, University of Regina, Regina, Saskatchewan, Canada

All plants in the image were modeled using L-systems (see article in this issue). The background sky was generated using the fractal algorithm of Saupe ("Quick recipe for random fractal clouds", *IRIS Universe*, Winter/Spring 1988). The depth of field was simulated using a modification of the algorithm by Potmesil and Chakravarty ("Synthetic image generation with a lens and aperture camera model", *ACM Transactions on Graphics*, Vol. 1, No. 2, 1982). Different viewing planes (the background lilac twigs, the apple twig and the daisies) were defocused separately, producing intermediate files which included transparency information. The intermediate images were subsequently composited using the alpha channel approach developed by Porter and Duff ("Compositing digital images", *Computer Graphics*, Vol. 18, No. 3, 1984).

back cover pic: meadow island
©1988 by Benjamin Garlick
Computer Graphics Research Laboratory
Department of Electrical Engineering and Computer Science
University of California, Davis

Contrary to reality, conventional fractal mountains will not shed rainfall. This island was created by dropping a few thousand "macro raindrops" that eroded random sinusoidal terrain. Ray intersection with the island was computed by marching into a spline of grid altitudes. Mandelbrot set clouds float above this lonely island.

contents

garden of L	4
orthopaedic implants	8
creatures of the complex plane	12
omnimax animation	16
vendor forum	20
software exchange	22
IRIS community	26
customer support	32
programmers corner	36

IRIS UNIVERSE ©1988

text & images © 1988 by individual authors.

editors:
Frank Dietrich
Zsuzsanna Molnar

design: Laslo Vespremi
digital images: Paul Haeberli
digital color separations: ImageSet, San Francisco

Silicon Graphics Inc.
2011 N. Shoreline Blvd.
Mountain View, CA 94039-7311
(415) 962-3320
<user@sgi.com>
!ucbvax!sgi!user

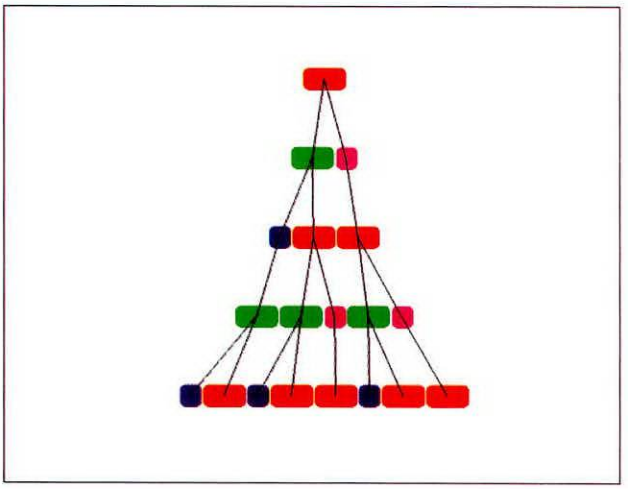


figure 1. development of a filament *(Anabaena catenula)*

garden of L

Przemyslaw Prusinkiewicz, Dave Fracchia,
Jim Hanan, Debbie Fowler
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada S4S 0A2

An attempt to re-create the structures and processes of life is immensely appealing to the imagination, balancing on the frontier between science and fiction. Known results [Langton 1988] range from the design of self-reproducing genetic messages to the replication of social behavior of animals. Between these extremes lie models of self-assembling viruses and simulations of multicellular organism development. As in physically-based modeling, where an astonishing degree of realism has been obtained by adhering to the fundamental laws of physics, realistic models of living structures can be produced by referring to biological mechanisms of development. The challenge of biologically-based modeling is reinforced by the fact that biological laws are often expressed qualitatively, without providing enough detail to construct a definitive model. Consequently, an attempt to simulate reality brings an insight into the nature of the underlying biological processes and structures.

In this article we outline our recent results related to the modeling of plants and simulation of their development. Many of these results were obtained in cooperation with Professor Lindenmayer and his students from the Theoretical Biology Group at the University of Utrecht. Technical details are presented in a paper by Prusinkiewicz, Lindenmayer and Hanan [1988].

As the primary tool for the modeling of plants, we employ the formalism of L-systems, originated by Lindenmayer [1968] as a mathematical description of plant development on the cellular level, and introduced to computer graphics by Smith [1984]. This formalism represents a plant structure using a sequence of symbols, each corresponding to a plant component or "module". In the modeling of lower plants, such as bacteria, algae, or fungi, symbols usually represent individual cells. In the case of higher plants, such as herbaceous plants and trees, cells are too low-level to serve as the basic modules. Consequently, the model is usually expressed in terms of structural components such as internodes (tree segments between two consecutive branching points), apices, and organs (leaves, flowers and fruits).

The process of producing new modules from existing ones is essential to biological development. On the microscopic scale, cells subdivide and produce new cells. On the larger scale, apices initiate new branches terminated with their own apices. The formalism of L-systems represents such processes using rewriting rules or productions. These terms, borrowed from formal language theory, denote rules for the substitution of string symbols. In contrast to the wider known Chomsky grammars which operate in a sequential manner, L-systems are inherently parallel. In other words, all symbols in a string are rewritten in a single step. This corresponds with the developmental processes in plants, where subdivision may occur simultaneously in many plant parts.

For example, let us consider the developmental simulation of a multicellular filament such as is found in the bacteria *Anabaena catenula* and various algae [Mitchison and Wilcox 1972, Lindenmayer 1982]. The symbols a and b represent cytological states of the cells (these states correspond with their sizes and readiness to divide). The subscripts l and r indicate cell polarity, specifying the positions in which the daughter cells of type a and b will be produced. The development is governed by an L-system comprised of four rules:

$a_r \rightarrow a_l b_r$, $a_l \rightarrow b_l a_r$, $b_r \rightarrow a_r$ *and* $b_l \rightarrow a_l$

Starting from a single cell a_r, this L-system generates the following sequence of strings:

a_r
$a_l b_r$
$b_l a_r a_r$
$a_l a_l b_r a_l b_r$
$b_l a_r b_l a_r a_r b_l a_r a_r$
...

Under a microscope, the cells appear as cylinders of various lengths. The a-type cells are longer than the b-type cells. The corresponding schematic image of filament development is shown in Figure 1.

In spite of its simplicity, the above example illustrates two important features of L-systems. First, the formalism is capable of amplifying the data base [Smith 1984], i.e. producing large, complex structures from very concise descriptions. Secondly, this amplification occurs by simulating the developmental processes which yield particular structures in nature. Thus, a model constructed with biological processes in mind can be directly applied to show entire developmental sequences. This point is illustrated by Figures 2-4.

Figure 2 represents four developmental stages of a leaf of the moss *Phascum cuspidatum*. *Phascum* leaves consist of single layers of cells, which makes them particularly suitable for detailed analysis under a microscope. An extension of L-systems from strings to graphs with cycles was introduced by Lindenmayer and Rozenberg [1979] under the name of map L-systems, and first applied to automatically plot cellular structures by de Does and Lindenmayer [1983]. The model shown in Figure 2 was generated using a map L-system devised by Martin de Boer.

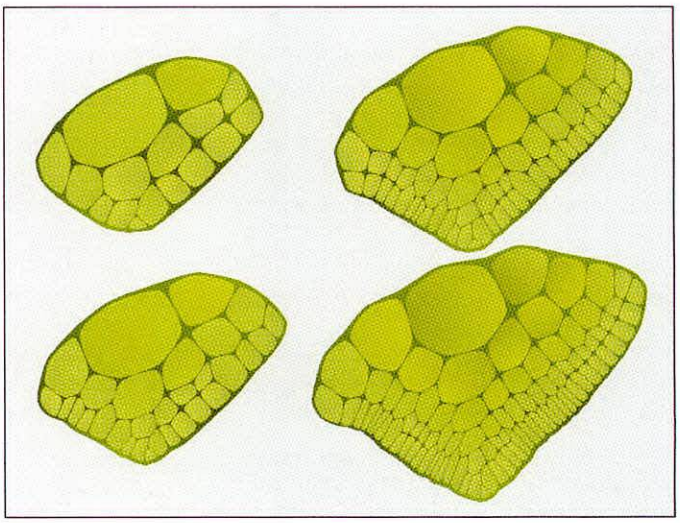
Figure 3 shows the development of a flowering plant, *Lychnis coronaria*. Its growth pattern was analyzed and expressed mathematically using a formalism similar to L-systems by Robinson [1986]. Figure 4 shows the development of *Mycelis muralis*, analyzed and formalized by Janssen and Lindenmayer [1987]. The flowering sequence of *Mycelis* is quite complicated from the viewpoint of control mechanisms employed by the plant. This is exhibited by the basipetal flowering sequence, i.e. progression of the blooming zone from the topmost parts of the plant towards its base. Since the oldest branches are near the plant base and the youngest ones are near the top, the basipetal sequence is counter-intuitive: the younger a branch is, the sooner it will produce flowers! In living plants such effects are controlled by hormones which flow across the branches and time subordinate processes. Accordingly, simulated flow of three hormones is the key to the *Mycelis* model.

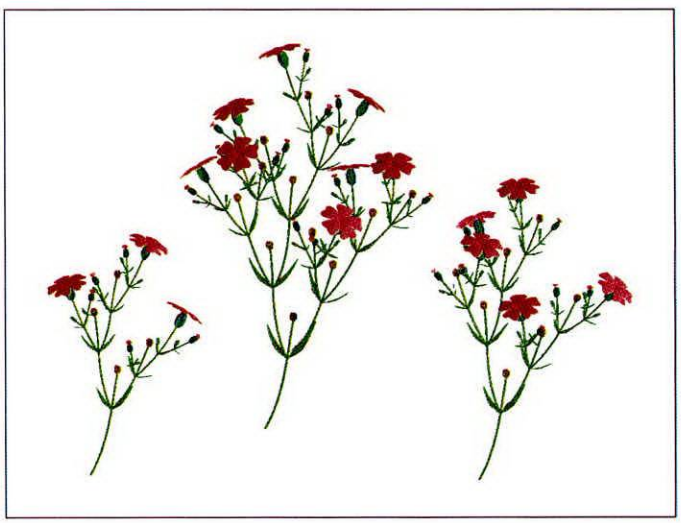
To present the link between biological reality and model construction in more detail, let us consider the example of a synthetic image "The Garden of L" (see cover). The central object of the scene is a lilac inflorescence. Its model incorporates the following rules, based on the analysis of real inflorescences:

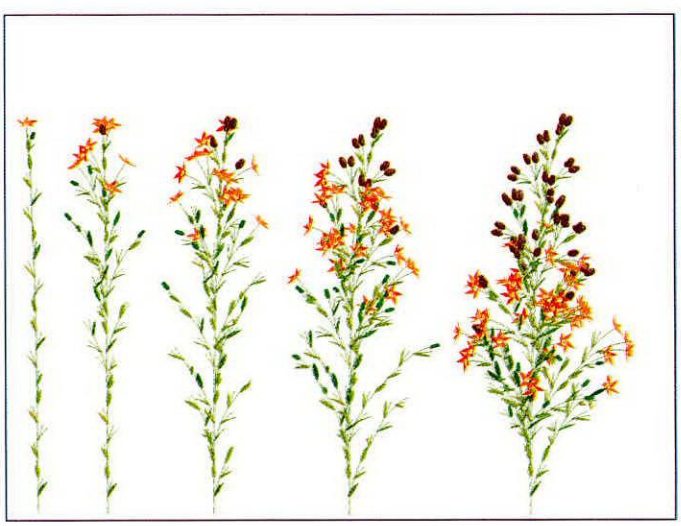
1. topology

a) The inflorescence has a recursive structure, with lower branches approximately replicating the higher portion of the inflorescence.

b) The first 2-8 lateral branches, counting from the







figures 2-4: *Phascum cuspidatum, Lychnis coronaria, Mycelis muralis*

apex of the mother branch, consist of an internode terminated by a single flower. The subsequent lateral (child) branches carry second-order (grandchild) branches. The number of second-order branches carried by a lateral branch increases linearly while traveling along a mother branch axis from the apex towards the base.

2. branching angles

a) Lateral branches occur in pairs. The axes of the consecutive pairs lie in planes perpendicular to each other.

b) The branching angle between the axes of the mother and the child branch is equal to 60 degrees.

c) If *A* is a mother branch, *B* is its child branch, and *C* and *D* are two grandchildren closest to the base of *B*, then the plane containing the axes of branches *C* and *D* is perpendicular to the plane defined by the axes of branches *A* and *B*.

3. proportions

a) All flowers within an inflorescence are approximately in the same developmental phase and have the same size.

b) The lengths of consecutive internodes along any axis form an arithmetic series, with the longest internode at the axis base.

c) The mother axis internode and the lateral internodes originating at a given branching point all have approximately the same length.

The information about proportions is the most difficult to infer, not only in the case of lilac, but generally in all plants. In contrast to the branching angles, which are fairly constant within a given plant species, the internode lengths are strongly influenced by the environment in which the plant develops. Consequently, the general rules governing proportions between plant components are masked by individual variations. A statistical analysis of a large number of inflorescences could reveal the underlying rules, but would be very time-consuming (and deadly to the flora around the graphics lab). In practice, it is simpler to assume the existence of a functional relationship between the lengths of the consecutive segments and their ordering numbers, and experimentally choose a function yielding correct proportions of the entire structure. In the case of the lilac inflorescence, the correct proportions were obtained assuming a linear function.

According to the developmental approach to modeling, the target plant structure results from a simulated growth process. In the case of lilac inflorescences, the consecutive segments of each axis as well as the lateral branches are produced exclusively by the apex (this type of development is known as subapical growth). As the internodes elongate in time, their lengths gradually increase while progressing from young internodes near the apex towards the older ones situated near the base. Thus, the spatial structure of an inflorescence results from the timing of the underlying developmental processes.

The branching structure of internodes forms the inflorescence skeleton, bearing the individual flowers. A single flower is small compared to the size of the entire inflorescence and can be adequately modeled by a collection of five bicubic surfaces which represent the four petals and the calyx. In contrast, the foreground lilac leaves in the Garden of L result from developmental processes. The midrib and the lateral "veins" form branching structures on which the blades are built. The overall leaf shape results from the sequence of angles under which the veins meet the midrib, and the relationship between their respective growth rates.

An interesting problem is related to the positioning of leaves on twigs. In principle, the leaves follow a pattern similar to that of lateral branches in inflorescences. They occur in pairs, with the stems of the consecutive pairs lying in planes perpendicular to each other and branch from the twig at an angle of 60 degrees. However, external factors, namely gravity and sun light, play an essential role in positioning the leaves so that they hang down and have blades facing the sun. The combined effect of gravity (negative geotropism) and sunlight is also responsible for curving the twigs up. These effects can be observed in the background lilac twigs. The specimen-to-specimen variation between the twigs was simulated by incorporating stochastic mechanisms into L-systems.

Compared to the lilac inflorescences, the apple twigs appearing in the upper left corner of the Garden of L and detailed in Figure 5 have a relatively simple structure. The short lateral branches leave the main axis at equal intervals. Each branch carries six leaves and six flowers. Essential to the model is the spiral arrangement of lateral branches along the main axis, and leaves encircling each branch. In both cases, the angle between consecutively placed elements (branches of leaves) is close to 137.5 degrees. Stevens [1974] provides the following intuitive explanation:

"Why that special angle? Because it has something to do with plant tissues? No. The angle only sets the points in proper relation to one another, so that each point makes a small angle with the older point in the previous whorl and a large angle with the younger point. That relation comes about quite naturally in the meristem of the plant as a direct result of each stalk's fitting the gaps of other stalks. ... The plant uses no mathematics, it simply grows the stalks where they have the most room."

The angle of 137.5 degrees also occurs in the daisies which grow in the lower right corner of the Garden of L; a magnified view of the flower capitulum is shown in Figure 6. The spiral arrangement of the florets results from the following formula, attributed to Erickson:

$\partial = n*137.5°$ $\quad r = C\ \sqrt{n})$

where:

- ∂ is the angle between a reference direction and the position vector of the n-th floret in the polar coordinate system originating at the capitulum center, and

- r is the distance between the capitulum center and the center of the n-th floret, given a constant scaling parameter.

The square-root relationship between the distance r and the floret ordering number n has a geometric explanation. Assuming that all florets have the same size and are densely packed, the total number of florets which fit inside a disk of radius r is proportional to the disk area. Thus, the ordering number n of the most extremely positioned floret in the capitula is proportional to r^2, or $r = C\sqrt{(n)}$.

Let us abstract from the details of the Garden of L to look at plant modeling from a larger perspective. What are the implications of the biologically-based approach? In the practical domain, plant models are useful in at least two areas. On one hand, simulated plants can be incorporated into computer-generated images of architectural projects, thus increasing their degree of realism. The developmental nature of the model makes it possible to anticipate landscape changes over time. On the other hand, computer simulations offer an unprecedented tool for the quantitative analysis of developmental mechanisms in plants. Real-life processes which occur in the time scale of months or years can be speeded up to seconds or minutes during computer simulations. Critical parameters, such as hormone propagation rates, can be easily modified. Hypothetical control mechanisms can be implemented and their implications visualized.

The realistic rendering of developmental models creates an illusion of life. It is easy to forget the underlying mathematical formalism and simply see plants growing, self-replicating, responding to external factors, even mutating. Although they are mere computer processes, we are fascinated by their life-like appearance and behavior.

references

de Does, M., and Lindenmayer, A. [1983]: "Algorithms for the generation and drawing of maps representing cell clones". In: H. Ehrig, M. Nagl and G. Rozenberg (Eds.): *Graph Grammars and their Application to Computer Science; Second International Workshop*, Lecture Notes in Computer Science **153**, pp. 39-57, Springer-Verlag, Berlin.

Janssen, J. M., and Lindenmayer, A. [1987]: "Models for the control of branch positions and flowering sequences of capitula in *Mycelis muralis* (L.) Dumont (Compositae)." *New Phytologist* **105**, pp. 191-220.

Langton, C. [1988] (Ed.): *Artificial Life: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*. Addison Wesley Longman Publishing Group (to appear).

Lindenmayer, A. [1968]: "Mathematical models for cellular interaction in development, Parts I and II". *Journal of Theoretical Biology* **18**, pp. 280-315.

Lindenmayer, A. [1982]: "Developmental algorithms: Lineage versus interactive control mechanisms". In: S. Subtelny and P. B. Green (Eds.): *Developmental Order: Its Origin and Regulation*, pp. 219-245, Alan R. Liss, Inc., New York.

Lindenmayer, A., and Rozenberg, G. [1979]: "Parallel generation of maps: Developmental systems for cell layers". In: V. Claus, H. Ehrig and G. Rozenberg (Eds.): *Graph Grammars and their Application to Computer Science; First International Workshop*, Lecture Notes in Computer Science **73**, pp. 301-316, Springer-Verlag, Berlin.

Mitchison, G. J., and Wilcox, M. [1972]: "Rules governing cell division in *Anabaena*". *Nature* **239**, pp. 110-111.

Prusinkiewicz, P., Lindenmayer, A., and Hanan, J. [1988]: "Developmental models of herbaceous plants for computer imagery purposes". To appear in the Proceedings of *SIGGRAPH '88*.

Robinson, D. F. [1986]: "A notation for the growth of inflorescences". *New Phytologist* **103**, pp. 587-596.

Smith, A. R. [1984]: "Plants, fractals, and formal languages". *Computer Graphics* **18**, No. 3, pp. 1-10.

Stevens, P. S. [1974]: *Patterns in Nature*. Little, Brown and Co., Boston.

figure 5: an apple twig

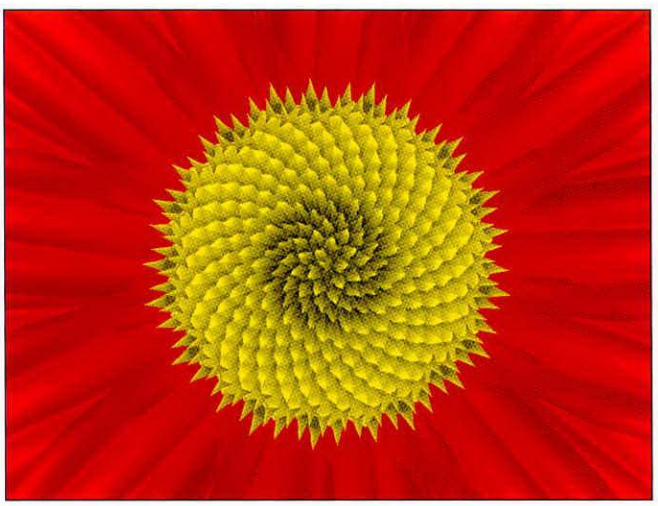


figure 6: capitulum of the painted daisy

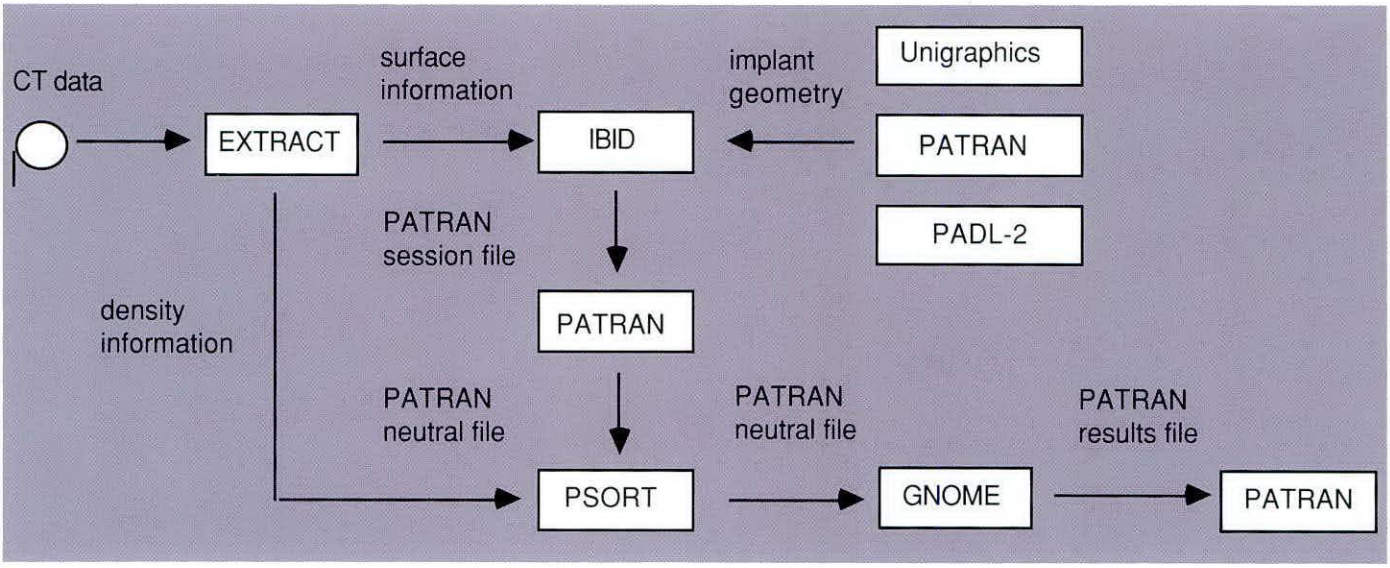


figure 1

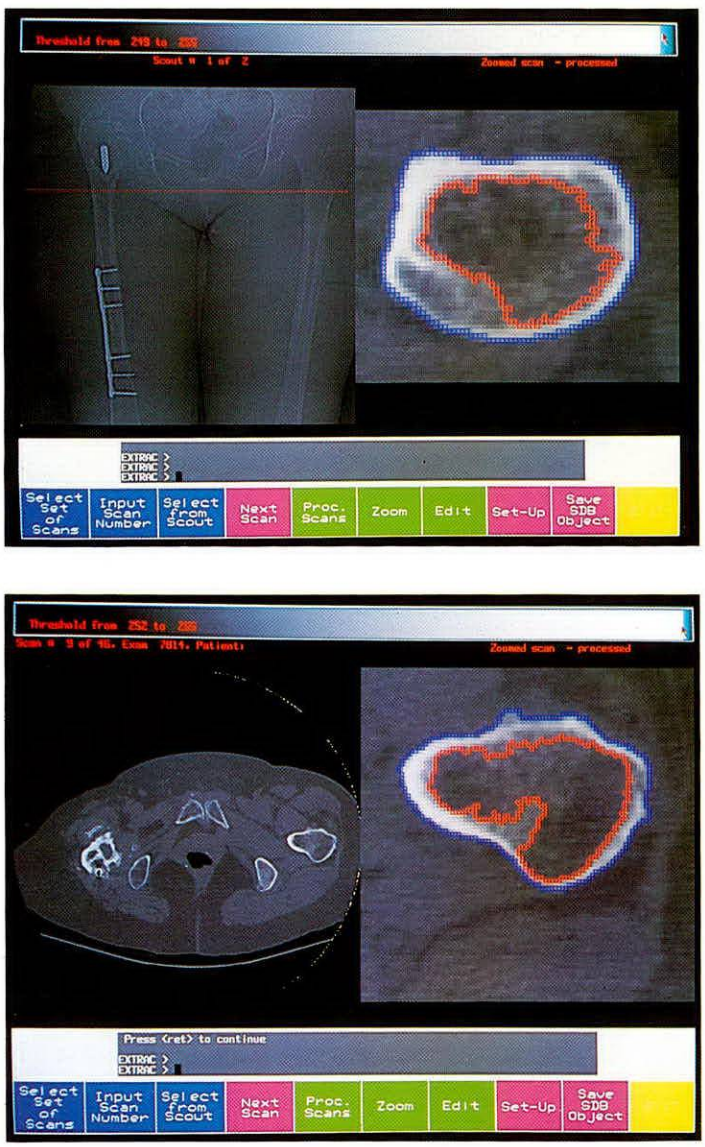
analysis and design of orthopaedic implants

Michael Piplani
HSS Program in Biomechanical Engineering
i219 Upson Hall
Cornell University
Ithaca, NY 14853
(607) 255-0990
<piplani@tcgould.tn.cornell.edu>

The study of stress states in bones, implants, and bone-implant systems has become increasingly important to the field of orthopaedics as the number of implanted total joint replacements and bone fixation devices has increased. In the United States alone over one quarter of a million total joint replacements are implanted annually. The long-term performance of these orthopaedic implants depends upon the structural integrity of the composite system which consists of the implant, the bone, and the biological or artificial interface layers between the implant and bone. It is insufficient to analyze these components individually since they are part of a composite structure. A change in one component will effect the load transfer and stress distributions in the other component. The Cornell-Hospital for Special Surgery Program in Biomechanical Engineering has assembled several advanced computational tools for the modeling and analysis of musculosketal systems and for the analysis and design of bone-implant systems. These tools are drawn from the areas of modelling, continuum mechanics, finite element analysis, medical imaging, geometrical representation, and computer graphics (Fig. 1). These tools will help researchers and designers prevent and predict failure in bone, implants, and bone-implant systems, and will enhance the understanding of the structural consequences of diseases such as osteoporosis (the bone loses calcium) and avascular necrosis (the bone experiences a loss in its blood supply).

Interactive computer graphics has been advantageous to the field of orthopaedics as it has to other fields because it has allowed the researcher to visualize data that could not be comprehended otherwise and to model skeletal systems, saving time and money. The quantities of data from a computerized tomography scan or nuclear magnetic resonance scan are too great to comprehend as lists of numbers. With computers these data can be visualized and organized effectively by constructing a manipulatable three-dimensional model of a bone from sets of two-dimensional cross sectional scans. The results from a complex three-dimensional finite element model stress analysis are also difficult to comprehend by just looking at plots of numbers. The capability to view these results on the model are made possible by interactive computer graphics.

By using the finite element method to model the structural aspects of these systems interactive design procedures become feasible to design implants. An implant can be designed, analyzed as part of the bone-implant system , and then redesigned until the optimum design is reached. This computer testing can be used to minimize the amount of physical and animal experimentation required in the overall design process. The finite element method can also be used to explain clinical observations such as failure of devices, loosening of the device in the bone, breaking or cracking of the device, or damage to the bone.



Ct scans, scout view, and zoomed regions can be displayed. Outer and inner femoral surfaces are delineated.

The first step in doing an analysis of a bone-implant system is to create a geometric model of the bone with EXTRACT. This is usually done by finding the boundaries of the bone to be modeled in the set of computerized tomography (CT) scans that cover the region of the body where the bone lies. Given all the boundary representations of a bone from a CT exam, these boundaries can be 'stacked' in the order the bone was scanned, and polygons can be lofted between each scan level. This yields a geometric model of the bone surface. Previously, and still to a large extent today, physicians and engineers usually evaluate computerized tomography scans by viewing black and white films consisting of sets of 4" x 4" pictures representing each scan. In creating a bone model using these films, the bone boundaries would be digitized by placing the film on a light tablet and tracing around the image of the bone with the stylus as carefully as possible. The process of hand digitizing can introduce error into the geometric model, and is slow and tedious for the person performing the task. Software has been developed to reconstruct the geometry of the bone from computerized tomography (CT) data, and to display and enhance this information. This software uses interactively controlled pseudo-colors to clarify and enhance displayed images. The user may zoom in on regions of interest, in a particular scan and may use any combination of a number of standard digital image processing filters to enhance the image. These filters include gradient, Laplacian smoothing, edge preserving smoothing, and two-dimensional fourier transforms with frequency domain filtering.

There are times when the CT image of the bone does not give as definite a boundary as the researcher would like. A filter such as an edge preserving smoothing filter when applied to the region of interest can give a more distinct boundary. Sometimes normal image processing techniques cannot separate individual structures in the CT image. In such situations the user may edit the image (add or delete pixels) to separate structures. When the user has a representation of the bone that is adequate for modeling, the boundary of that bone may be found automatically (the computer finds the boundaries of that bone slice) or semi-automatically (the user specifies where the boundary starts). Error, once introduced into the process by hand digitizing, is eliminated, and this process is much faster. The outputs of this software are fed into IBID. Here the geometric information for the bone surface and the radiographic density information for each voxel within the bone is constructed. The density information may be used later to assign material properties to the bone for structural analysis using the finite element method.

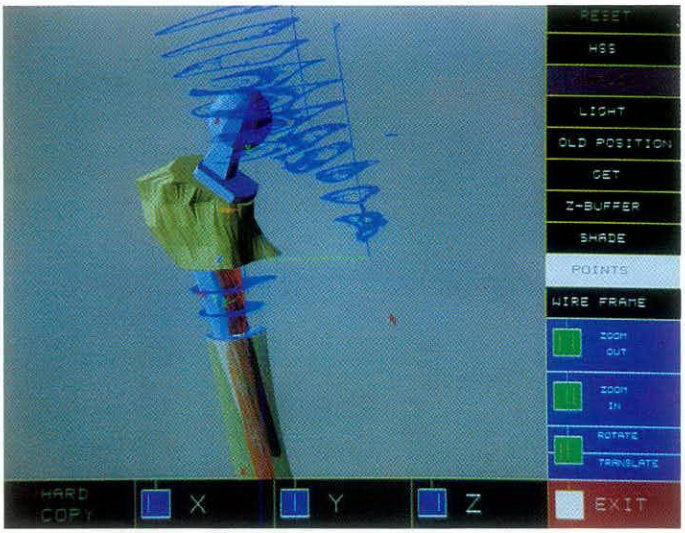
The next step in the process is to make a model of the implant to be tested. The implant can be created in one of several solid modeling packages, such as: PATRAN, PADL-2, or CATIA. Once this has been done the geometry of the implant is integrated with the geometric data base of the bone model.

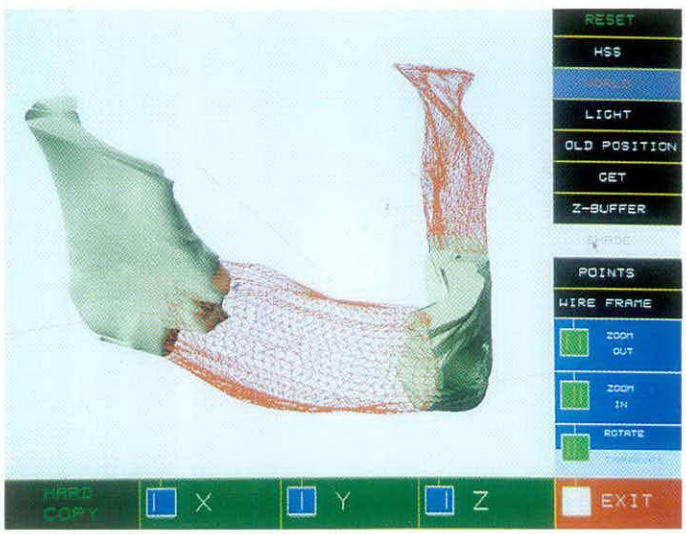
Biomechanical engineers and orthopedic surgeons have similar requirements for manipulating three-dimensional structural data. These include the ability to produce accurate, aesthetically pleasing, transparent color shaded images of multiple objects; the ability to extract geometric information such as dis-

tances, thickness, direction of surface normals; the ability to provide relative motion between objects; the ability to section objects; detect collisions between objects; and the ability to interface with CAD/CAM packages for finite element analysis and numerically controlled machining tasks. A geometric data base using a polygonal boundary representation, has been developed to meet these needs. Given a bone and implant in the geometric data base, the next step is to use PSORT to position the prosthesis with respect to the bone, and to prepare the combined structure for finite element model generation.

Placement of the prosthesis in the bone is critical to the success of the model. Since the typical bone has a medullary cavity, it is interesting to note that for positioning an implant within a bone the identification of the inner surface may be more important than the accurate representation of the outer surface. The implant is typically positioned within the inner surface. Depending on the design, it may be press-fit or cemented into the bone. In either case, the bone will usually be reamed out to accommodate the implant. Proper sizing of the implant with respect to the bone is an important consideration in design and surgical planning. The geometric database's capabilities have been exploited to aid in the positioning process. The outer bone surface is made transparent to allow the user to see the prosthesis within the bone. Cross sections of the bone-implant system may be taken in any orientation to see how the implant has been placed with respect to the inner and outer bone surfaces. A portion of the bone can be removed to simulate the actual surgical procedure. For example, in a total hip replacement the head of the femur is resected, and in a total knee replacement the tibial plateau and the condyles of the distal femur are removed. When the implant has been positioned satisfactorily, the resulting bone-implant geometry can transferred to a finite element mesh generator.

When the resulting bone-implant geometry is in a finite element preprocessing package, the model is prepared for analysis. Loads simulating forces in the body due to the body's weight or ligaments attached to the bone are placed on the model. When the researcher is satisfied with the resulting model, the analysis is run in a finite element package. Depending on the size and complexity of the model the job may be run on a micro-computer or the Cornell National Super Computer. The resulting data is then viewed using a finite element post-processor and interpreted by the researcher on the IRIS workstation.





top: a prothesis in a femur with head resected.

bottom: a mandible (jaw bone) displayed using Gouraud shaded polygons.

High performance graphic workstations have made these visualization techniques possible. The right combination of software tools has enhanced the design and production of orthopaedic implants. The goals are to provide better implants and better patient care. As the technology advances, the ease of use should improve. It will be possible to place all of the software in one workstation to be used by surgeons in preoperative planning and in post-operative evaluation of implant performance.

Acknowledgement: I would like to thank Professors Dean Taylor and Donald Bartel for their encouragement and assistance in writing this article.

CDI TECHNOLOGIES, INC.

ARE YOU LOOKING FOR A CAD SYSTEM?

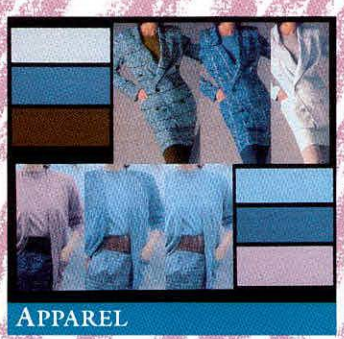
MANY CLAIMS ARE BEING MADE.
FOR AN IN-DEPTH COMPARISON, INSIST ON PLACING
ANY COMPETITOR'S SYSTEM PHYSICALLY NEXT TO OURS:

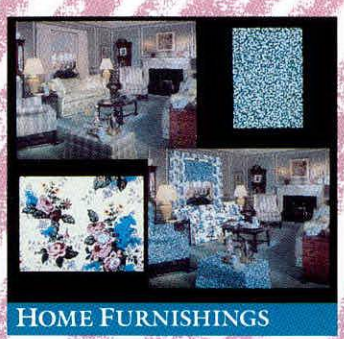
COMPARE **PRICE**
Our products range from the least expensive, a PC based system, to the most expensive 3D design to manufacturing system with automatic flattening of the pattern pieces.

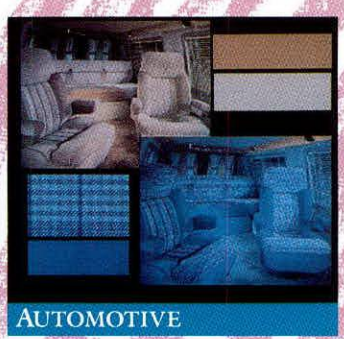
COMPARE **FUNCTIONALITY**
CDI offers more functions over a wider product range than any company. From fabric design, knitting, weaving, to styling, to manufacturing.

COMPARE **QUALITY**
Our software provides the latest mathematical techniques to insure clear, realistic images using all 16 million shades of color. All of our software was written by us under the direction of designers and manufacturers such as yourselves.

COMPARE **SPEED**
Our PC based system and our high end 3D system offer greater speed than any competitors. Important? You bet! Performs more functions quickly for a fast ROI.


APPAREL


HOME FURNISHINGS


AUTOMOTIVE

"We've looked at all the systems on the market and CDI has no competition. They definitely are the Cadillac of CAD systems."
***Norman A. Deitch**/Turquoise International Ltd.*

COMPUTER DESIGN, INC.

5270 Northland Dr.
Grand Rapids, MI 49505
616 361 1139

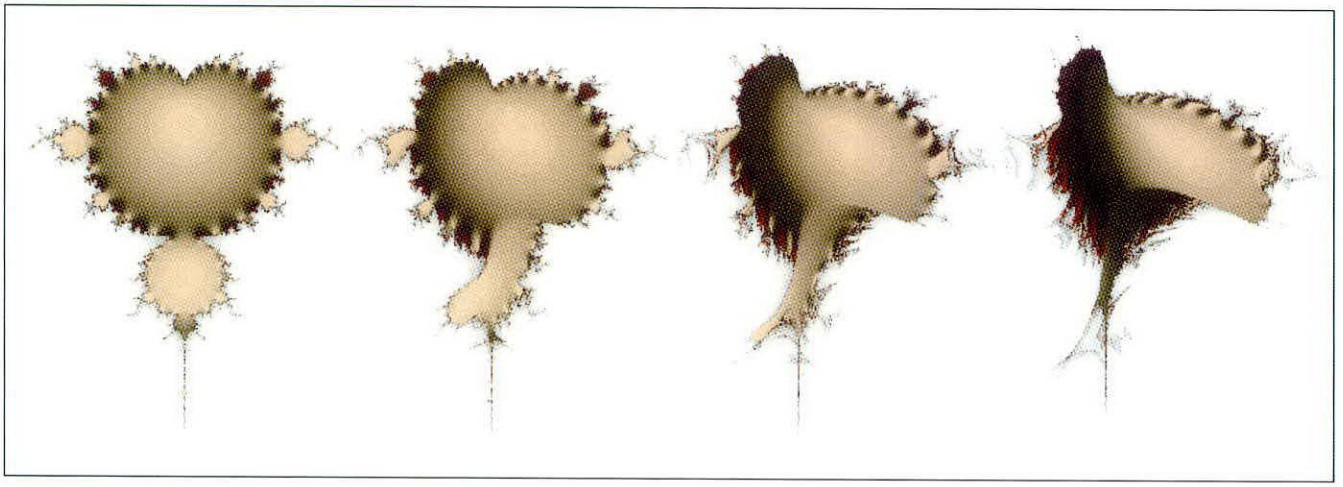
1466 Broadway
Suite 1507
New York, N.Y. 10036
212 575 2060

Henrietta House
5th Floor
9 Henrietta Place
London England
W1A 45N
1 629 3271

BrandTex
Nordlunde J1
DK 7730 Brande
Brande, Denmark
45 7 181111

Nixdorf
Via Piranesi 46
20137 Milano
Italy
39 2 73961

CIM Textile
Lilienthalstrasse 19
D-4800 Bielefeld 11
Germany
521-86011



creatures of the complex plane

Jeffrey Ventrella
Advanced Graphics Research Lab
005 Heroy Bldg.
Syracuse University
Syracuse, NY 13244-1206
<JJVENTRE@SUNRISE.ACS.SYR.EDU>

The images illustrated here represent a zoology of forms which are the products of specially designed functions. These functions may be thought of as stylized sculptor's chisels which are used to carve into the dense fabric of the complex plane. The mathematical process known as "iteration in the complex plane" has recently become a popular topic due to some striking discoveries. The most celebrated is the Mandelbrot set, a geometrical object which has an infinity of detail and a highly organic form. Dr. Benoit Mandelbrot, the founder of fractal geometry, likened the iterations of his complex functions to the act of sculpting. The more times the function is iterated the more detail can be carved out of the plane. These images are some of my own interpretations of the poetic qualities latent in the Mandelbrot set. While the qualities are essentially mathematical attributes brought about by discrete operations, they are treated here as metaphorical.

Like many other fractal artists who carve the complex plane, I am interested in the symbolic implications of this new world of visuals, a world which was not possible seven years ago. This, in addition to an inclination toward the surrealist movement in art, and a dedication to natural form and color, has contributed to the motivation behind these pictures.

abstract biogenetics

Aside from the sculptural aspect of iteration, fractal functions may be thought of as genetic codes determining the anatomies of the shapes. While exploring the inner-workings of a function and finding interesting alterations one may get a sense of unravelling the DNA helix. To me it is an artform of abstract biogenetics, with the IRIS serving as an elaborate petri dish. Fractalizer Peter Oppenheimer of the New York Institute of Technology and others promote such analogies of fractal art to biological creation. (1) For the artist working with fractals such as the Mandelbrot set, a new method of crafting visuals becomes apparent: the organization and beauty of the forms are not entirely creations of the artist. With practically no visual manipulation aside from the actual techniques of display, fractals seem to have a life of their own. But a computer artist may wish to inject a personal style into the forms to coincide with the already stunning imagery that exists by default. This may be done by "tweaking the parameters", either with clearly defined mathematical transformations or with more subjective, "amathematical" techniques.

the tool

These deformations of the Mandelbrot set are explained from the standpoint of a visual thinker concerned with the manipulation of pre-existing imagery. To begin, as stated, the Mandelbrot function is the mathematical tool for carving out this well-known fractal form. One may also think of any alteration of this function as a tool for generating other fractals. This has been done to an extent by mathematicians and artists interested in illustrating complex dynamics or exploring new fractal aesthetics. Thomas Papathomas and Bela Julesz of AT&T Bell Labs have animated deformations of the set as a part of their research in human visual perception of motion and depth. (2) The present approach is to a great

extent amathematical: the method of generating the images relies more on intuition and a visual sense than on any clear-cut mathematical concept. Still, a description of the logical starting point for this technique may help in describing how the images came about.

The Mandelbrot Set is defined as the set of all locations C in the complex plane in which Z remains finite during the recursive operation:

$Z_{n+1} = Z_n^2 + C$

where Z and C are complex numbers, and $Z_0 = 0$. C serves as a constant which represents a location in the complex plane being tested. In short, each C in a given window of the plane is used in the function, which is iterated a given number of times. The final state of Z after n iterations determines whether that location C is in the set or not. A more thorough description of the generation of the set may be found in Dewdney (1987, 1985), Peitgen and Richter (1986), and Mandelbrot (1983), among many others. Techniques for colorizing images of the set may be based on information such as the number of iterations it took Z to exceed a certain magnitude for each location, or the actual value of Z after n iterations. The former is a standard way of displaying the set but both are used in the present treatment, including other methods based on the nature of the series of Z's created by the process.

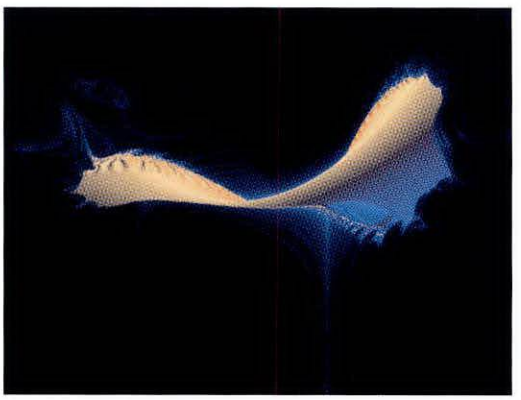
In my program, the key parameters used as genetic "tuning knobs" of the Mandelbrot set are actually real number variables in an expression of the function with its real and imaginary parts separated:

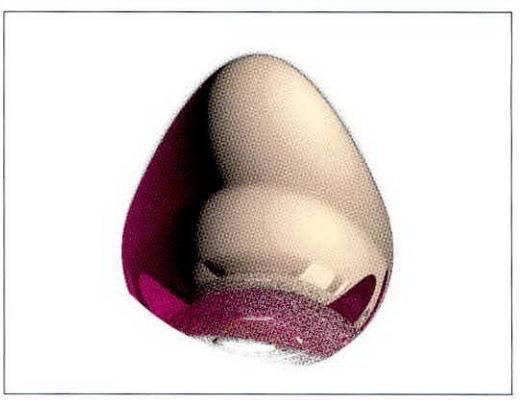
$Zrn+1 = Zin2 - Zrn2 + Cr$
$Zin+1 = 2\,Zrn\ Zin + Ci$

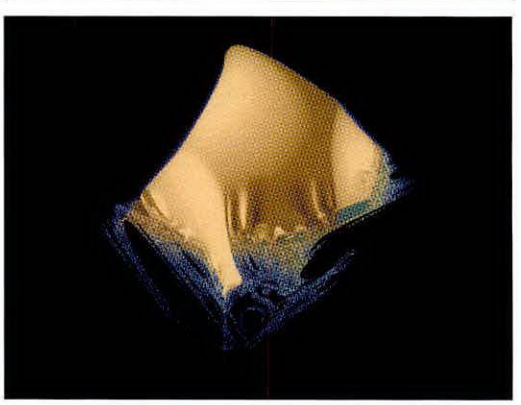
The first of the Mandelbrot set's "freaky others" I discovered was the result of changing the 2 in the second line to -2. This seemingly small change transforms the set into a creature having a distinct 3-fold symmetry and exhibiting complexity only at its extremities. One of the first discoveries made: it is possible to generate shapes in which the fractal parts (the complex, fuzzy areas) are shifted to occupy distinct localized areas or configurations.

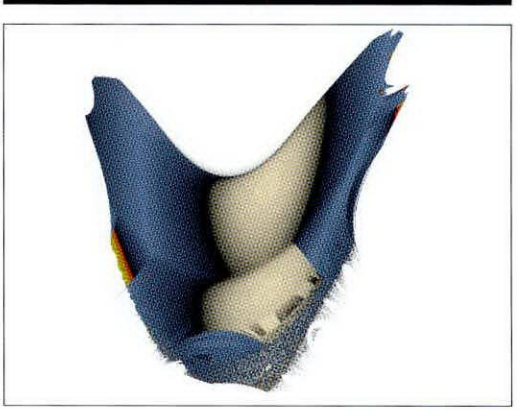
More varieties of manipulation are made possible by attaching tiny tuning knobs to the eight variables in the example. For instance, the Zr could be expanded to Zr+t, or Zr/t, etc. with t being any small fraction entered interactively. I'll leave the description open-ended at this point since any further variations one might try would be subject to a particular programming or artistic style.

In experimenting with these genetic tuning knobs, you might soon get a sense of the Mandelbrot Set as a piece of mathematical rubber, but a unique kind of rubber subject to the properties of the complex plane. To me, the complex plane is a rich ground for poetic exploration. The mathematical explanations behind most of the









poetry came only as an afterthought. And for this reason, the math was revealed in a more beautiful and metaphorical way than it could have been otherwise. This is the way some people like to learn mathematics.

two and a half dimensions

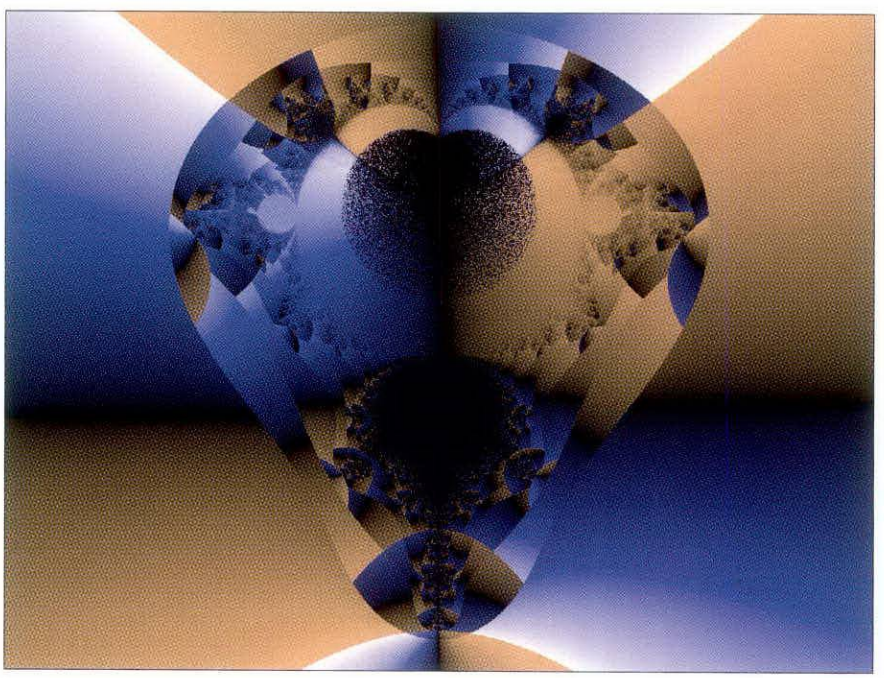
The third dimension is the IRIS workstation's specialty. It is designed to be proficient at performing transformations in 3D, and fast. While superworkstations like the IRIS are bringing the three dimensions of Euclid down to a hardware level, non-euclidean geometries such as fractals are giving rise to new methods of image-making that don't necessarily follow the rules of standard geometry. Fractal objects can have dimensions which lie between whole numbers.

These images suggest illusionism and depth, but they do not involve any use of IRIS 3D geometry, as do the beautiful landscapes by Dietmar Saupe. (see last issue of the IRIS Universe). This illusionism is suggested by utilizing the smooth gradations created by mapping these functions. The gradations can be interpreted as topographies of three-dimensional surfaces, as D. Saupe has done. It is provocative to compare the Z in these mappings (especially as existing inside the set) to the Z-axis of three-dimensional space. But in these examples, the analogy is expressed as pure color. I prefer to keep this analogy loose, for the same reason that painters of the Modernist tradition often provoke a tension between two-dimensionality and illusion.

In actual practice these pictures are generated on a two-dimensional cartesian plane, but in appearance, and also in theory, the dimensions are quite fuzzy. You could say that these images are examples of two-and-a-half dimensional computer art.

technique

Although no use of IRIS 3D functions were used, the whole technique would not have been made possible without the sheer display speed and interactivity the IRIS offers. This kind of technique requires constant visual evaluation of parameter inputs. Given control over time-consuming factors like picture resolution and the number of function iterations, a workable image can be generated in a matter of seconds. After literally hundreds of trials and changes involving tweaking parameters, running new colormaps, and adjusting pans and zooms, things like resolution and iterations can then be jacked-up for the final running of the image. It is then ready for the camera (a 35mm camera on tripod, with low ASA print film, and shutter speed set at about one second).



What this discussion may illuminate for those already familiar with fractal art is a divergent way of approaching the Mandelbrot set. Most other treatments involve magnifying very remote areas of its boundary and then applying wild colorations to the image. This approach does not interpret the Mandelbrot set literally. It begins with the set as a biological creature, with a genetic code, a physiology, even a personality, to be manipulated (with mathematical intrigue but not rigor), and then displayed in its entirety using naturalistic colors.

the advanced graphics research lab at syracuse university

This work was produced on an IRIS 3020 at the Advanced Graphics Research Lab at SU, an all-campus research facility with an emphasis on image processing and data visualization. There have been recent developments in networking graphics applications, via fiberoptic cables, to the university's new Northeast Parallel Architectures Center, located on campus. Interfacing the IRIS with NPAC's Connection Machine will shortly open up new possibilities for further and deeper explorations with fractals in the complex plane.

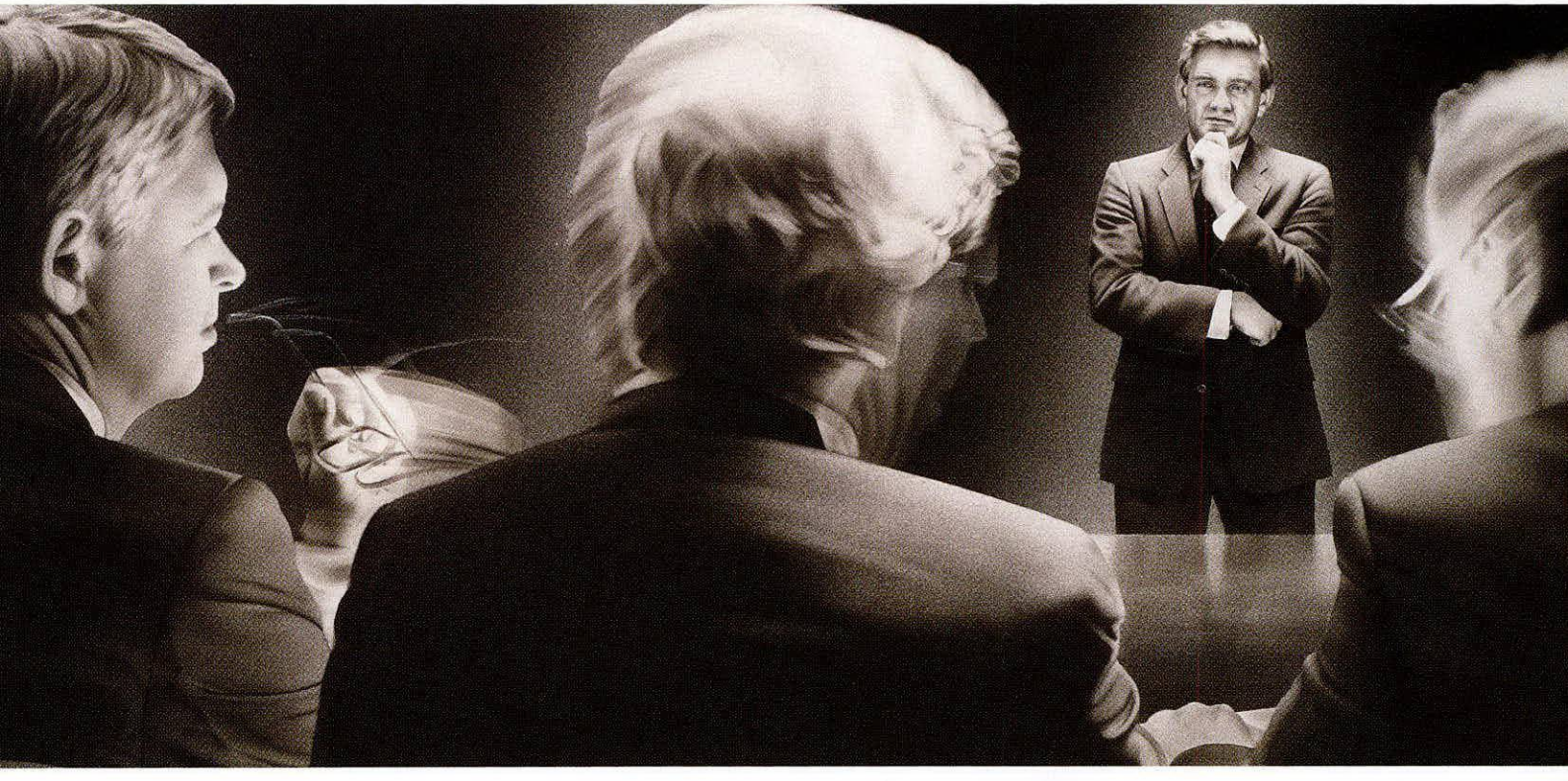
footnotes

1. Oppenheimer, Peter, from a presentation at Siggraph '87 and a paper in the course notes: "The Modeling of Natural Phenomena", page 55

2. Papathomas & Julesz, "Animation with fractals from variations on the Mandelbrot set", The Visual Computer 1987, pages 23-25

The deformations of the set described in this article were done by varying the exponent in the function.

You've just been given six months to design a new product. Now, all you have to do is figure out how.



They just handed you an impossible deadline. And now you're going to have to turn your whole engineering team upside down to meet it.

Or, you could turn to I-DEAS™. The solids-based mechanical computer-aided engineering (MCAE) system that helps you deliver superior designs, on time and on budget.

With I-DEAS, you can accurately simulate a product's performance long before creating detailed drawings or building expensive prototypes. That's because, unlike traditional CAD/CAM systems, I-DEAS is based on advanced, 3-D solid-modeling technology.

And, I-DEAS provides a complete and unambiguous product definition for design, analysis, testing, drafting, and manufacturing. So your team can visualize and evaluate a product early in the design process, and at all points downstream.

Design automation software from SDRC helped Aerojet Electro Systems cut product development time by 28%.

SDRC has over 5,500 I-DEAS systems in operation around the world, each supported by our 20 years of engineering experience. We also have a team of consulting engineers who are available to provide customized services to meet your individual design requirements. It is this unique combination of technology, engineering experience and consulting expertise that will make the most of your investment in MCAE.

If deadlines are putting you in an uncomfortable position, call us at 513-576-2003. Or, write: SDRC, 2000 Eastman Drive, Milford, Ohio 45150.

SDRC

© 1988 Structural Dynamics Research Corporation.

computer animation for the omnimax film *seasons*

Richard Weinberg, Ph.D.
University of Southern California,
Vibeke Sorensen
California Institute of the Arts

introduction

Seasons is a full length OMNIMAX film for which we produced almost two minutes of computer animation. The film presents the four seasons with glorious images and describes the struggle to understand what causes the seasons to change. *Seasons* is a tour de force of special effects in OMNIMAX format, including time lapse, macrophotography, motion control and computer animation. The computer animation in *Seasons* is an integral part of the film; a powerful tool in the filmmaker's toolkit.

OMNIMAX is the largest motion picture film format in use today. Each 70mm, 15 perforation frame of OMNIMAX film is ten times larger than a standard frame of 35mm motion picture film. This film can be projected either on a dome (OMNIMAX format) or on a flat screen (IMAX format). The excitement of working in OMNIMAX format compelled us to participate in the project.

We wanted to use computer animation to explain the changing seasons in two views of the universe: the Ptolemaic and Copernican systems. In the early Ptolemaic view, the earth was the stationary center of the universe; the sun and the stars orbitted the earth attached to transparent crystal spheres. By attributing a peculiar orbit to the sun, early astronomers were able to account for the changing seasons. In the current Copernican view of the universe, the earth orbits the sun, and the tilt of the earth on its axis causes the changing seasons.

omnimax

Seasons is not the first use of computer animation in OMNIMAX format. Nelson Max, a computer scientist and mathematician at Lawrence Livermore National Laboratories, pioneered the technique developing the mathematical transformations to map a flat piece of film onto a curved dome.

The ACM/SIGGRAPH '84 conference, in cooperation with the Science Museum of Minnesota and several corporate sponsors, produced a computer generated OMNIMAX short subject in a mammoth cooperative effort among more than a dozen organizations. The film is now distributed under the name *The Magic Egg*.

We cut our teeth on OMNIMAX computer animation for SIGGRAPH '84, with Richard initiating the film, and Vibeke producing a sequence of animation depicting the constellations of the Zodiac in wireframe.

Remembering how demanding the format is, we boldly volunteered for the *Seasons* project, and split the work between our laboratories, with CalArts doing the design and USC the production film recording.

seasons

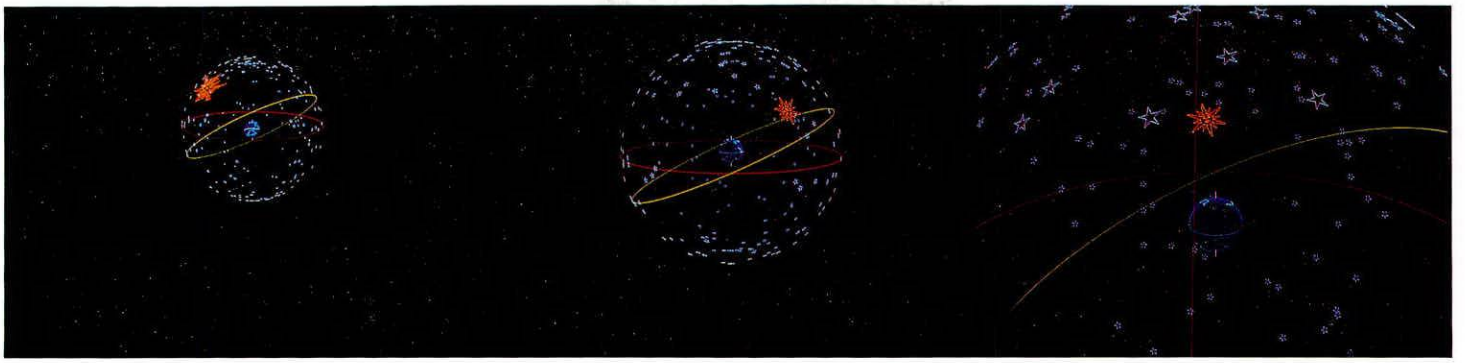
We decided to use vector style animation instead of raster animation for several reasons. We felt that we could achieve the effect of suspending the solar system in space, and illustrating the two views of the universe effectively using only vectors. From the SIGGRAPH '84 ONMIMAX film, we learned that vector animation in ONMIMAX is quite striking. Brightly colored vectors on a black background are very effective when projected on the dome, and the high resolution film recorder provided vectors without any jaggies. And of course, the computing involved with vector animation is orders of magnitude less than fully rendered color raster graphics.

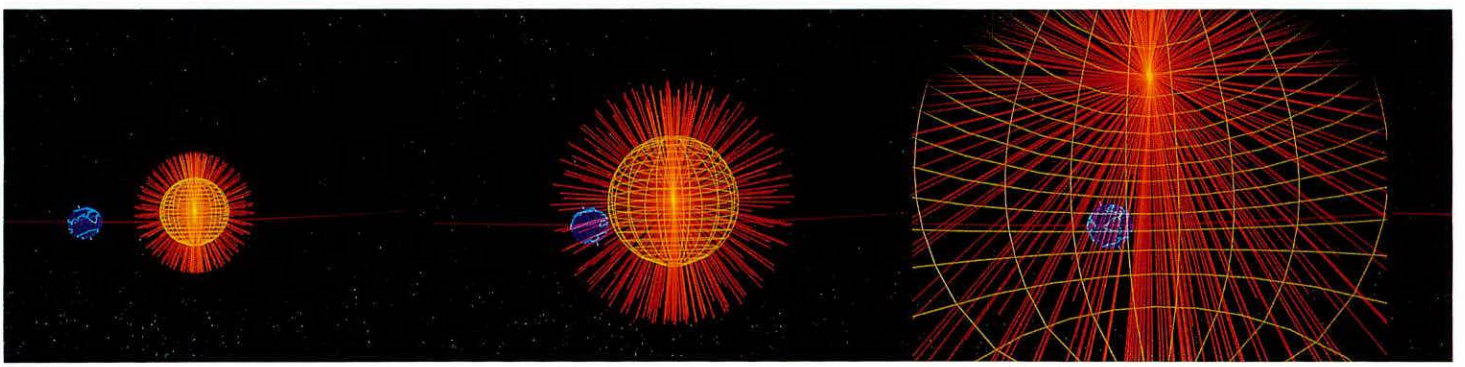
design

Ben Shedd, the producer/director, and Vibeke Sorensen researched various models of the solar system and how to represent them on film so that Vibeke could develop the storyboard. The storyboard begins with a live-action sequence of the carnival rides of the Minnesota State Fair, ablaze with neon lights. The look is similar in style to wireframe animation, and provides an ideal transition to the computer animation. During a two moths period, Vibeke designed the objects and their motions using the Silicon Graphics IRIS.

CalArts Computer Animation Lab is equipped with a Silicon Graphics IRIS 3020 running Wavefront Software, including *Model* (for building solid models), *Preview* (for setting up animation sequences and previewing them in real time), and *Image* (for rendering the animation frames). The IRIS is well suited to real time animation, which allowes us to preview movements in real time and to refine the choreography. It also allows us to view our objects in color, which helps isolate various parts of models and scene compositions so that we can work more quickly and carefully with complex models and movement. Both of these capabilities were essential to the creation of the computer animation in *Seasons*.

Vibeke created models of the sun with the Wavefront Model program. The





continents and star data had been previously obtained from NASA through Caltech for use in the Constellation sequence in *The Magic Egg*. The star positions are actual star data, including positions in space and relative brightnesses. We used the brightest 500 stars in the list for the *Seasons*.

One problem encountered in animating the motion of the earth around the sun is the speed of the earth's rotation on its axis. We had to slow down its apparent rotation to show more than one orbit around the sun in a reasonable amount of screen time, while still giving the impression that the earth is revolving about its axis without strobing. A twelve-day year proved appropriate. In 15 seconds of screen time, the earth orbits the sun once and revolves about its own axis 12 times. The animation shows how the constant 22.5 degree tilt of the earth's axis accounts for the changing seasons.

When Ben Shedd came to CalArts to view the work in progress, he would place his eyes about two inches from the IRIS screen and watch the entire sequence, simulating the omni dome. One could get sea-sick from too much tilting and swerving on the Omnimax dome, so movement through space had to be gentle.

One feature of computer animation which particularly intrigued Ben was the ability to have everything in the frame in focus at once, and infinite depth of field. (In this case, from one side of the universe to the other!) We wanted to create the feeling of being in outer space, so we had to be careful not to remind the audience of the screen's borders. We carefully composed the scenes to avoid hitting the frame line in the animation, especially at the bottom of the screen. Objects entered and exited at the periphery of our eyes, from the sides and tops of the screen.

Moving objects need to be carefully choreographed for the large screen. If the motion between frames is too great, the objects will *strobe*, or jump from frame to frame as they move across the large screen. In order to alleviate strobing as the sun orbits the earth, we copied a classic animation technique and used *blur streaks*, lines trailing a moving object. Blur streaks give the impression of smooth motion, and are much simpler and faster than computing true motion blur.

Composition for the large screen is quite different than for, say, television. In OMNIMAX, if an object moves all the way across the screen, the audience must turn their heads to follow it, once they have noticed it. In portraying the two views of the universe, we kept the center of attention near the center of the screen, although the audience actually flies right through the sun at the end. The horizon line in OMNIMAX and IMAX is quite low on the screen; if objects are too high on the screen, the audience must tilt their head backwards to see them.

Initial tests were done on 3/4" videotape directly from the Silicon Graphics IRIS workstation. After a number of weeks, we were ready to shoot a low-resolution test (in 35mm!). We stepped through the entire sequence a frame at a time, filming directly from the IRIS 3020 monitor. The quality of the image impressed us all, being crisp and clear in color. The resulting animation was edited into a working version of the film for Ben to view, in order to finalize timing, composition, narration and music. Once all of the design work was complete, then the data was transferred (via floppy disk!) to Richard's laboratory at USC.

film recording and production

Producing computer animation in OMNIMAX format presents a unique challenge. Foremost is the high resolution required for the large screen. To write the images onto film we developed a film recording system based on the Dicomed D48C film recorder at USC, with a 65mm OMNIMAX camera mounted upon it. We designed and built an IBM PC/AT interface to control the film recorder. This interface is a board which connects the 16 bit PC/AT bus to the Dicomed 16 bit internal interface. Since the interface is a one-of-a-kind item, we had to develop the control software for it, using Turbo Pascal. The film advance mechanism of OMNIMAX camera itself was controlled through a serial port of the IBM PC/AT.

We used both an IBM PC/AT and a PC/RT for the animation. The PC/RT was used to interpret the Wavefront channel data describing the motions, and produce a group of 4 x 4 matrices for each object, for each frame. These matrices were then transferred (again by floppy disk) to the PC/AT. The PC/AT was used to transform the vectors making up the sun, the earth and the stars by the 4 x 4 matrices, clip the vectors to the screen, clip off the far side of the earth, darken the dark side of the earth, and output the vectors to the film recorder. As the sun obits the earth in the Ptolemaic system (and vice versa in the Copernican system) the side of the earth nearest the sun is bright, and the opposite side is darkened.

We had originally planned to use the OMNIMAX transformations developed by Nelson Max, but instead simply used a wide angle lens simulation to cut down on computing and testing time. The final footage looks very good in both OMNIMAX and IMAX formats. Since the objects we portray are mostly circular or spherical and do not have long straight lines, an exact OMNIMAX distortion was not critical.

The positions of the stars in the Ptolemaic scene were derived from a database of stars supplied by Caltech. The stars were represented as five pointed stars situated on a sphere centered on the earth, based on the drawings of early astronomers. In order to make the crystal sphere of stars look three-dimensional, we used depth cueing, so that the closer stars appeared brighter. Two of the three scenes required a stationary star background. For this we decided to optically combine a painted star field with the vector animation. First the star field art work was shot, and then the exposed film was loaded into the camers on the film recorder for the computer graphics.

The computer animation follows a time-lapse sequence of the Minnesota State Fair Midway at night. We dissolve from a stylized neon sun-face on a rotation carnival ride to the sun-face in the computer animation. This dissolve required matching the position and rotation rate of the computer generated sun to the live-action sun. This section of the film is a dramatic transition from the bustling sped-up state-fair, to the peaceful reaches of outer space where the computer animation takes place.

Our 'low-resolution' test animation was shot in 35mm. A miniature dome at Graphic Films was used to preview the 35mm film. by sticking our heads right inside the dome, and being careful not to get our hair caught in the projector, we could simulate the peripheral vision effects of viewing the OMNIMAX film in a full size dome.

The final animation was produced on 15 perforation 65mm film. Each frame required about two minutes of computing and film recording, and each vector was drawn several times to get enough brightness on the film. The longest single computing/recording run lasted 37 hours. Cutting and splicing OMNIMAX film accurately is very difficult, so each animated sequence needed to be filmed in its entirety in a single pass. We used Kodak 5247 motion picture film for the production.

The location of USC is ideal for working in this format. MetroColor labs, essentially the only film lab equipped to handle the format, is a half hour from USC, and the Mitsubishi IMAX theater is literally across the street from USC. We were able to take exposed film to the lab by midnight and get a 70mm OMNIMAX workprint back the next morning, which we screened between the regularly scheduled films at the IMAX theater that same day. This rapid turnaround was a tremendous asset in the tight schedule we faced.

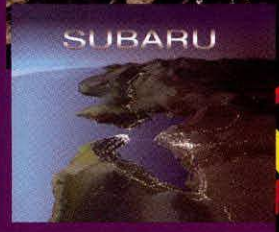
conclusion

Seasons premiered at the Science Museum of Minnesota during the summer solstice of 1987, and is currently showing in several Omnitheaters around the United States. The film was sponsored by the Science Museum of Minnesota, and produced and directed by Academy Award winner Ben Shedd. The film is narrated by William Shatner, and features Vivaldi's *Four Seasons* as performed by the St. Paul Chamber Orchestra. The soundtrack for the computer animation was composed and performed by Michael Stearns. Using the videotape test footage, he was able to compose the musical score for the computer animation. In six track stereo, the sound moves around the theater as if in orbit with the earth, adding an exciting dimension to the animation and reinforcing the motion.

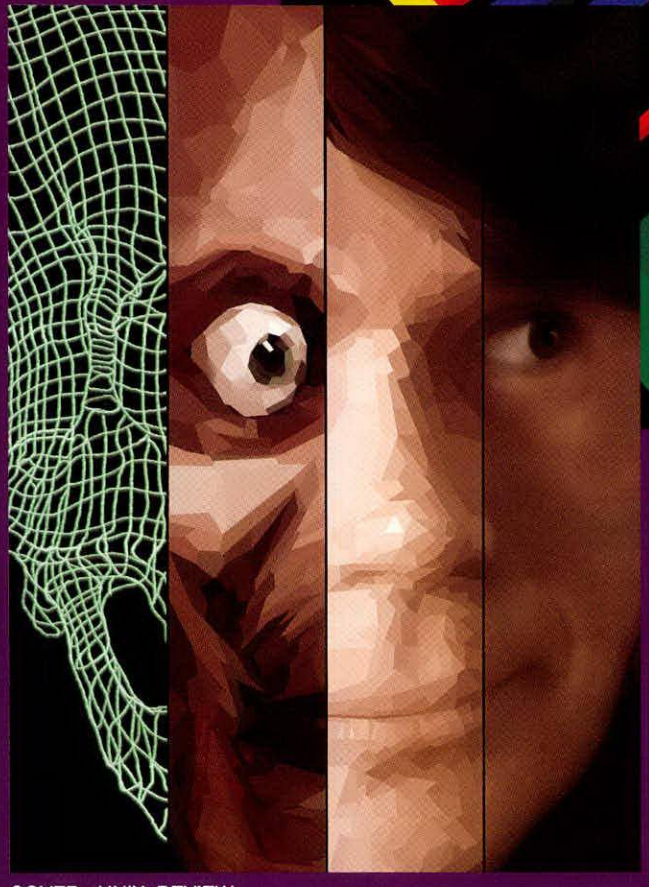
Computer animation allowed us to present an understandable and exciting representation of two opposing views of the universe for the OMNIMAX film *Seasons*.

EIDOLON

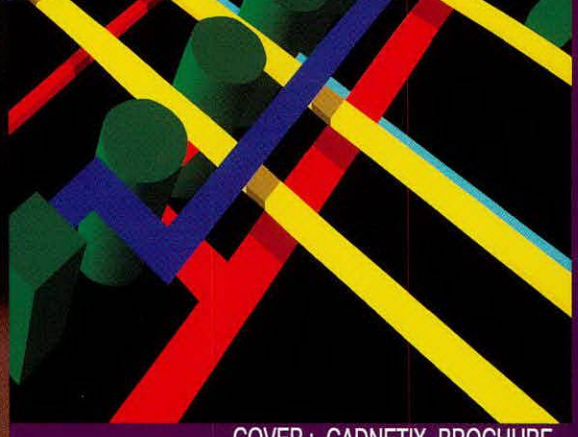
DIGITAL GRAPHIC ART NOW ™



BACKGROUNDS: TV ADS
FOR: PRODUCTION & ANIMATION ARTS



COVER: UNIX REVIEW



COVER: CADNETIX BROCHURE

DIGITAL RETOUCH & ILLUSTRATION

ENTIRE PAGE (15 MILLON PIXELS, 60 Mbyte) DELIVERED DIGITALLY TO SCITEX. DIGITAL TRANSFER & COLOR SEPS.: C G L & VISICOLOR DENVER

FOR:
- REALTIME PIPELINE IMAGE PROCESSING PERIPHERALS FOR IRIS WORKSTATIONS
- DIGITIZING FROM VIDEO AND UP TO 20,000 X 20,000 X 24 BITS FROM FILM
- FILM RECORDING TO DICOMED 4X5" & MITCHELL PIN REGISTERED 35mm MOTION

CALL: GW HANNAWAY & ASSOCIATES / EIDOLON (303) 440-9631 BOULDER, COLORADO

videotaping from an IRIS workstation

Robert P. Marcus, President
RGB Technology
Berkeley, California

Want to convert the output from an IRIS workstation to television (NTSC) video for video taping or transmission?

Historically, designers of high-end computer graphics displays diverged from the video standards used by broadcasters, T.V. cameras and video taperecorders in order to achieve the steadier images and higher resolutions preferred by engineers and other computer users. From a technical standpoint, there are many differences between standard composite video (NTSC) and the analog signal generated by a workstation:

. . . Standard video is an interlaced video system. This means that a video frame is comprised of two separate fields, one displaying the odd numbered video lines and the other, the even numbered lines, at a display rate of 60 fields (30 frames) per second. By comparison, the IRIS displays all video lines 60 times per second. This is known as a 60 Hz *noninterlaced* display, as compared to 30 Hz *interlaced* for composite video.

. . . The IRIS has an RGB display providing a screen resolution of 1024x768 or 1280x1024 pixels (picture elements). Standard composite video has only half the resolution in both the x and y direction.

. . . The IRIS display feeds red, green and blue signals to the monitor separately. Composite video encodes the three into a single signal. Composite television monitors contain the circuitry to decode this signal back into separate red, green and blue signals.

Whatever the advantages of computer displays in terms of high resolution and steadier imagery, NTSC (television) video remains the most commonly used communications medium for animated visuals. The problem becomes one of how to convert the high resolution computer graphics information into a lower resolution video and to do so in a manner that does not lose critical information.

Simple NTSC video output options offered by workstation vendors are based on programmable frame buffers and tend to be somewhat unwieldy to use. For example, they require the user to redraw images at lower resolution within a predefined "video window" on the screen. This is inconvenient and may require software modification. Moreover, normal output to the system's high resolution RGB monitor is not possible in the NTSC output mode.

Enter the RGB/Videolink, an interface between the IRIS's high resolution display and virtually all television devices, including monitors, recorders and projectors. The RGB/Videolink takes a *full screen* color or grayscale input, performs line and pixel averaging, provides genlock, sync generation and encoding to output NTSC video. It accomplishes the conversion in real time and without processing burden or interruption to the host or its high resolution display.

The RGB/Videolink is the first device to meet the criteria for a state-of-the-art *scan converter*. All graphical information is averaged down and included in the lower resolution image. The conversion is accomplished in real time, i.e. complete conversion in less than one frame time. Finally, full color is provided for smooth shading. This means 8-bit digital-to-analog and analog-to-digital converters on each color channel, 24 bits in total.

Each RGB input from the IRIS is digitized, compressed and anti-aliased using line and pixel averaging techniques and stored in a compressed format. It is important to note that the RGB/Videolink does not drop lines of video, as was done by an earlier generation of scan converters. Instead, it averages down the information on the workstation screen in both the x and y direction. An attribute of the process of averaging down is that the picture is "anti-aliased," which contributes to picture quality. The digitized image is then converted by D/A's to RGB signals and encoded into NTSC (or European standard PAL) composite video. An RGB output at RS-170A video resolution is also provided.

The RGB/Videolink scan converts the full screen and not merely a quarter screen (525x525 pixel) "video window". This means that there are no software modifications required. Furthermore, the RGB/Videolink does not interfere with the normal operation of the host monitor. This supports simultaneous use of both the host monitor and video monitors or VCR's, always a nice feature and critical in certain applications.

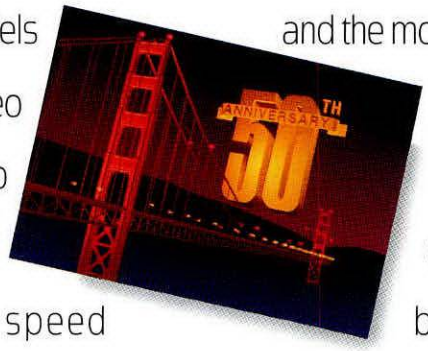
Beyond providing a technical solution to scan conversion, the RGB/Videolink is designed to be a complete solution to making effective video tape presentations.

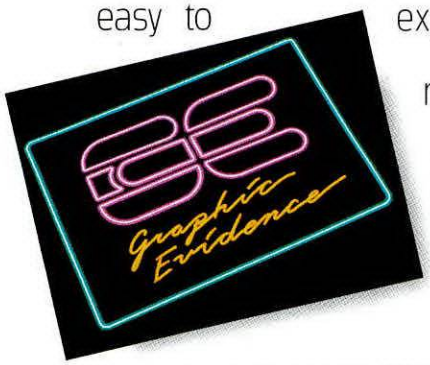
Want to mix workstation graphics with live video? An enhanced version fo the RGB/Videolink is equipped with a video mix option, to allow integration of live or pre-recorded video with the output from the IRIS. This capability may be used to put a background behind the computer generated image or to insert a video image, e.g. of a "presenter." The video mix is available in NTSC format.

With other standard features such as freeze frame, aspect ratio correction, color bars and video blanking, the RGB/Videolink is capable of generating effective videotape presentations without the need of ancillary equipment, video engineers or post production services. The result is easier, cheaper, and better videotape presentations, and faster turnaround.

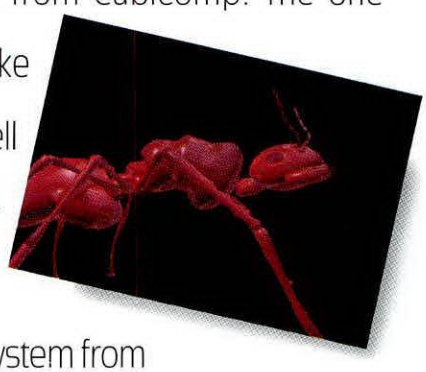
VERTIGO V2000™
FOR NEW DIMENSIONS IN 3D ANIMATION.

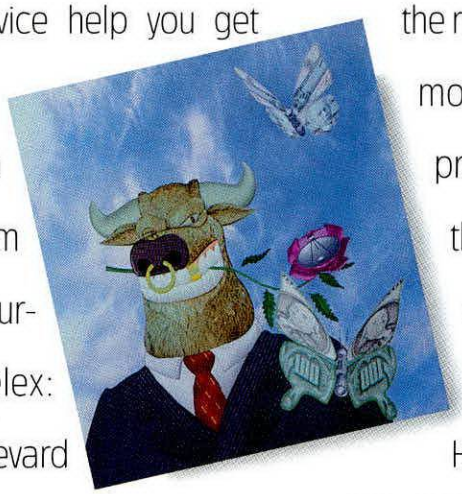
One system makes 3D animation look better than ever ■ The Vertigo V2000. The workstation-based system designed specifically to meet all your video animation needs ■ Everything is included. All the hardware. All the software. So you can produce complex models and the most intricate animation sequences for broadcast-quality film or video productions—quickly and easily. From digitizing artwork, to recording final frames ■ And when your production needs grow, it's easy to expand. You can add more speed by adding more render accelerator boards. You can add more render stations. And design stations ■ So we'll meet your needs today. And tomorrow ■





Now here's the best news. The V2000 is from Cubicomp. The one company that really knows how to make you a success. Because we sell more than just systems. We sell complete solutions. From planning and evaluation to training operators through system installation and checkout. Plus, ongoing support and service help you get the most out of your system from day one ■ What's more, Cubicomp offers the broadest line of video animation products, anywhere. Products that are based on feedback from the world's largest installed base of animation users ■ See for yourself. Call for our reel. 800/331-9678. (In CA 415/887-1300. Telex: 294295.) Or write Cubicomp Corporation, 21325 Cabot Boulevard Hayward, CA 94545 ■ And let us add an entirely new dimension to your 3D animation ■

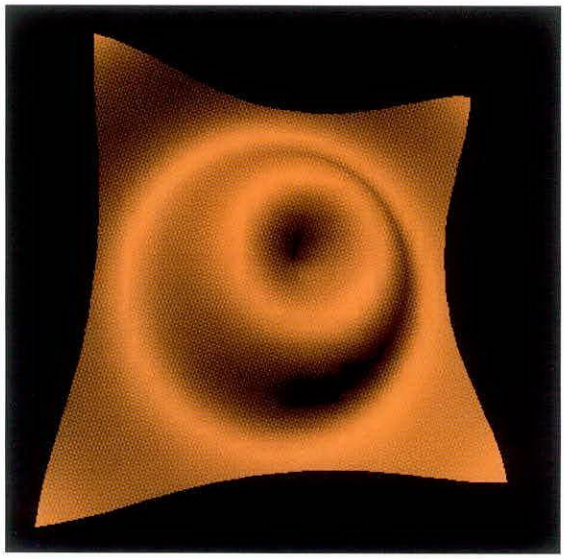




CUBICOMP CORPORATION

Picture the Possibilities ■

(1) LORIMAR LOGO: © 1987 Cubicomp Corporation, (2) GOLDEN GATE BRIDGE: © 1987 San Francisco Production Group, for KPIX, (3) GRAPHIC EVIDENCE LOGO: © 1987 Graphic Evidence, (4) ANT: © 1988 Graphic Evidence, (5) BULL: © 1987 Cubicomp Corporation, V2000 is a trademark of Cubicomp Corporation © 1988 Cubicomp Corp.



mathematica

Wolfram Research Inc.

Mathematica is an interactive software system that incorporates a high-level programming language. *Mathematica* does three basic kinds of computation: numerical, symbolic and graphical. Unlike an electronic calculator or a standard computer language, *Mathematica* works with numbers of arbitrary precision. As well as making possible many calculations of mathematical interest, this capability has the practical consequence of giving the user unprecedented control over numerical roundoff errors.

Built into *Mathematica* is a complete set of mathematical functions. Beyond the standard exponential and trigonometric functions, *Mathematica* includes Bessel functions, hypergeometric functions, elliptic integrals, and much more. Most of these functions were previously accessible only through incomplete books of tables. *Mathematica* incorporates efficient numerical algorithms for matrix manipulation, numerical integration and numerical equation solving.

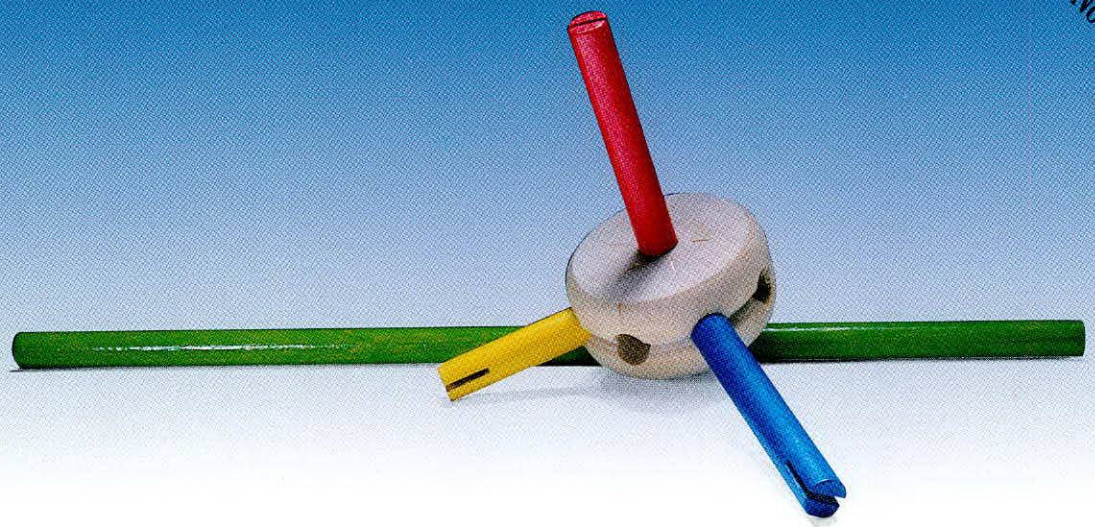
Beyond numerical computation, *Mathematica* can also work with algebraic formulae. Built into *Mathematica* are all the standard operations of algebra and calculus, including polynomial factorization, as well as symbolic integration and differentation. The ability to manipulate and derive exact algebraic formulae makes it possible to automate a wide range of calculations that could hitherto only be done by hand. With *Mathematica*, the user can get explicit formulas for the solutions to equations, rather than just finding numerical results. This makes a much wider range of calculations possible, and also allows the user to understand the complete structure of a solution.

The third major area covered by *Mathematica* is graphics. *Mathematica* can plot functions and data in two and three dimensions, in black and white or color, allowing the user to visualize results of calculations. *Mathematica* can also take symbolic descriptions of arbitrary geometrical objects, and produce three-dimensional color pictures. Thus, for example, the user could generate a list of faces in a complex polyhedron using *Mathematica*'s symbolic and numerical capabilities, and then produce a three-dimensional picture of the polyhedron using its graphical capabilities.

Mathematica uses the PostScript standard for all its graphics output. This resolution-independent form can be rendered on computer screens, laser printers, phototypesetters, or even on huge posters. *Mathematica* also can produce encapsulated PostScript, which can be included in desktop publishing systems. In addition to over 700 built-in functions, *Mathematica* incorporates a powerful programming language, which enables the user to add their own functions. The *Mathematica* language supports several forms of programming, including structured programming in the style of C or Pascal and functional programming in the style of APL. *Mathematica* also allows a novel rule-based style of programming in which the user specifies transformation rules for mathematical expressions which match a particular pattern. Such rules can be constructed by almost direct transcriptions of formulas from textbooks.

The high-level nature of the *Mathematica* language allows users to create complex programs easily. The primitive functions provided by *Mathematica* correspond directly to primitive operations in mathematics itself. On UNIX systems, *Mathematica* can interact with external programs through pipes. Thus, for example, *Mathematica* could take data from a laboratory instrument, or could communicate with an existing external numerical program.

In the area of graphics, *Mathematica* appears to be unique in providing a full implementation of three-dimensional hidden-surface graphics in resolution-independent PostScript. *Mathematica* is being ported to the IRIS 4D series.



Now on SGI IRIS 3000 Series and SGI 4D/60.

As a child, you mastered the skills necessary to use the most sophisticated graphics programming tool kit in the world.

Remember your childhood days. Building things was so simple.

A few basic tools and a little imaginative spark and you could accomplish just about anything.

If only building complex custom graphics applications was as easy.

With FIGARO it is. FIGARO is the enhanced PHIGS graphics programming tool kit so sophisticated, using it is almost child's play.

Serious tools designed for serious tinkering.

Programming custom graphics applications is serious business. Tinkering with each little detail takes time. The last thing you need is to have your graphics tool kit fall short. Causing bottlenecks. Or unexpected delays.

Another setback you'd prefer to avoid is having to start from scratch each time you make a change.

FIGARO gives you production power and design flexibility. A complete modeling system makes it easy to generate complex 3D models. Build an engine or create a molecular model. Then run the engine. Or fuse two molecules.

And FIGARO's hierarchical data base structure allows you to build inter-dependent components, yet edit and make changes independently. Selectively highlighting. Choosing multiple views. And displaying them simultaneously.

Edit your data base and FIGARO will update your image. Rapidly. With ease.

FIGARO offers unequaled performance and infinite freedom. Simple solutions. Serious possibilities.

Uncompromising across the board.

In the infantile days of graphics programming, device-independence meant compromised performance.

Well, the graphics tool kit has grown up.

FIGARO is designed to maximize your programming potential at no expense to performance or speed. Regardless of which major platform you're currently using.

As we've said before, FIGARO is designed so that getting up to speed with your platform requires minimal effort. But the true beauty here is, transferring that knowledge over to another platform doesn't require any.

Just like riding a bike.

Be it Hewlett Packard, Silicon Graphics, Sun Microsystem, Prime, Tektronix, Apollo, DEC or IBM, you need not make any code changes to operate effectively.

FIGARO has incorporated the extensions most asked for by sophisticated applications builders. So, with each individual platform, you get the exact same powerful tools. More than portability, universal application.

Together providing performance and portability, FIGARO stands alone.

Working from a higher standard.

FIGARO is just what you'd expect from Template Graphics Software. TGS is the company that wrote the book on the PHIGS standard.

Best of all, when you buy FIGARO, you've bought the full support of the industry leader, TGS. With training courses, reference manuals and a toll free hotline . . . to keep you up and running.

FIGARO. It's child's play to use.

FIGARO. Performance & portability with PHIGS.

For a FREE BROCHURE call 1-800-428-7588, in CA 1-800-824-4490, or send this coupon.

Name_____

Title_____

Address_____

City, State, Zip_____

Phone_____

Mail to: Template Graphic Software, 9685 Scranton Road, San Diego, CA 92121

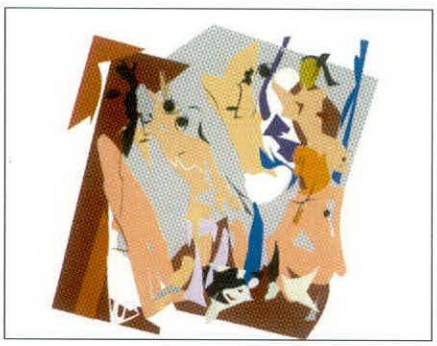
TGS TEMPLATE GRAPHICS SOFTWARE

A United Telecom Company

art dream

David Haxton
William Paterson College

The *Art Dream* is a fantasy journey through a series of twentieth century paintings. A modern art book opens and the paintings on each page become three dimensional. Objects in the paintings begin to move and the viewer in some cases moves through the individual paintings. In reality these paintings are two dimensional, seen only as illusions of three dimensional space. In the *Art Dream* the viewer sees a three dimensional world, through the use of computer generated models, derived from reproductions of the original paintings.

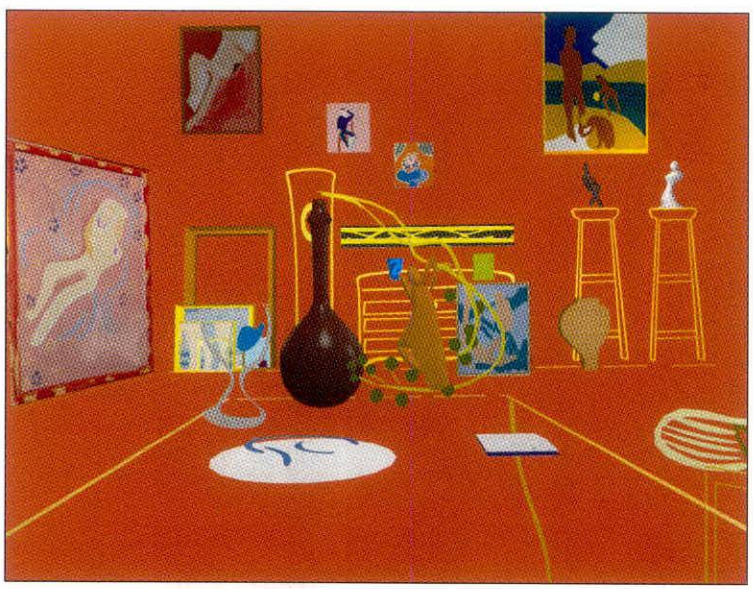


Picasso's *Women of Avignon*

Women of Avignon explodes

Picasso's *Three Musicians*

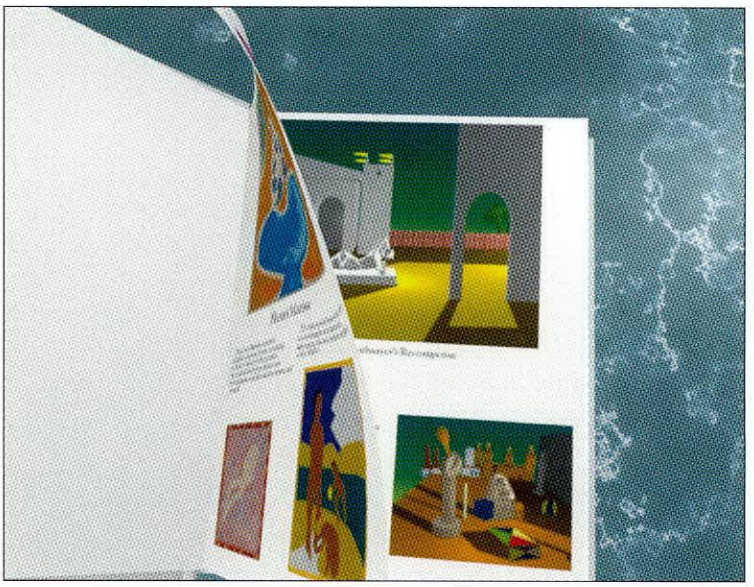
The art dream was created at the Center for Computer Art and Animation at William Paterson College (WPC). The piece was directed by David Haxton (director of the center and faculty member). The art direction was by Joseph Shingelo (technical director of the center). Janet Rentel (graduate assistant) was the production manager and animator. Cheryl Kelly (graduate assistant) was the chief animator for the project. Several students in the computer art program were engaged to produce models, animation and paintbox images. They include: Robin Biggio, Karen Blum, Ray Caramanna, Lori Herkomer, Renee Latham,

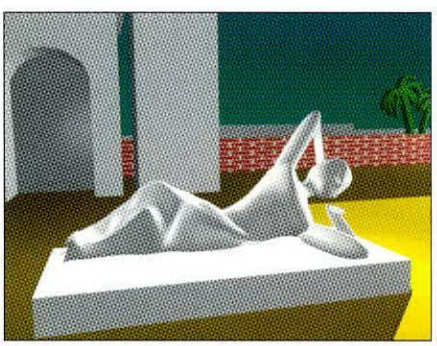


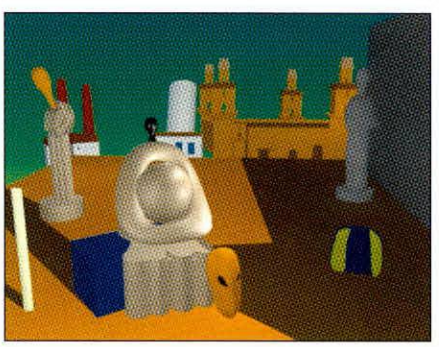
Matisse's *Red Room*

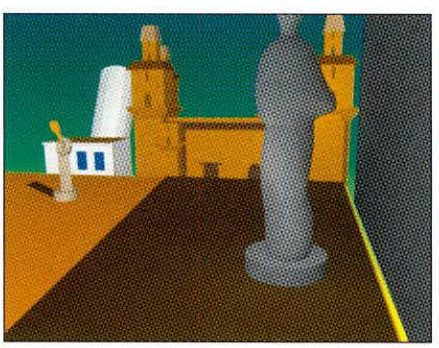
Chris Mackinnon, Cynthia Teichman and Dan Speck. The primary objective for the center is to create computer generated art. The fact that a work is computer generated is secondary to the conceptual development of idea. However the unique properties of the computer as an art medium are always considered as an aspect of the overall piece.

The Center for Computer Animation at WPC has been in operation for the past three years. The major objective of the Center is to foster an environment for the development of creative approaches to computer animation. The facility currently houses five IRIS 3130 computers and one IRIS 4D series compute server. The laboratory utilized Alias I software on all of the systems and Neo-Visual on one of the systems. The *Art Dream* is being produced entirely with Alias I software.

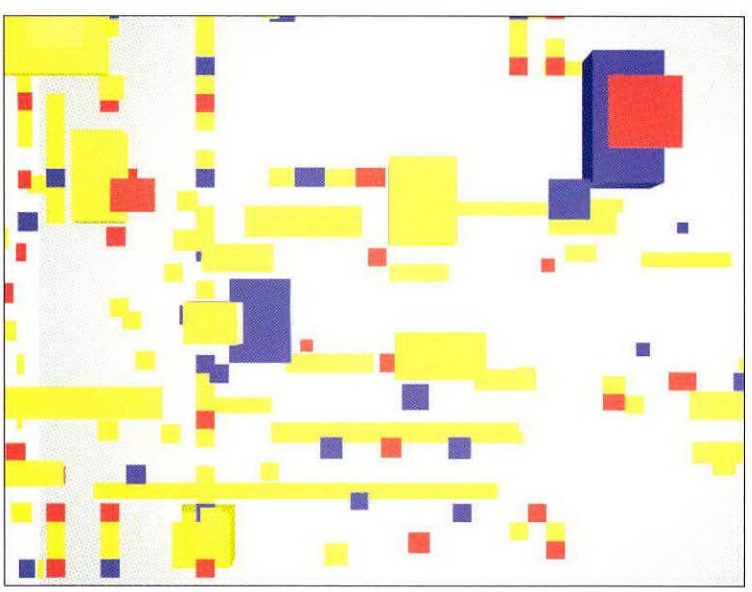






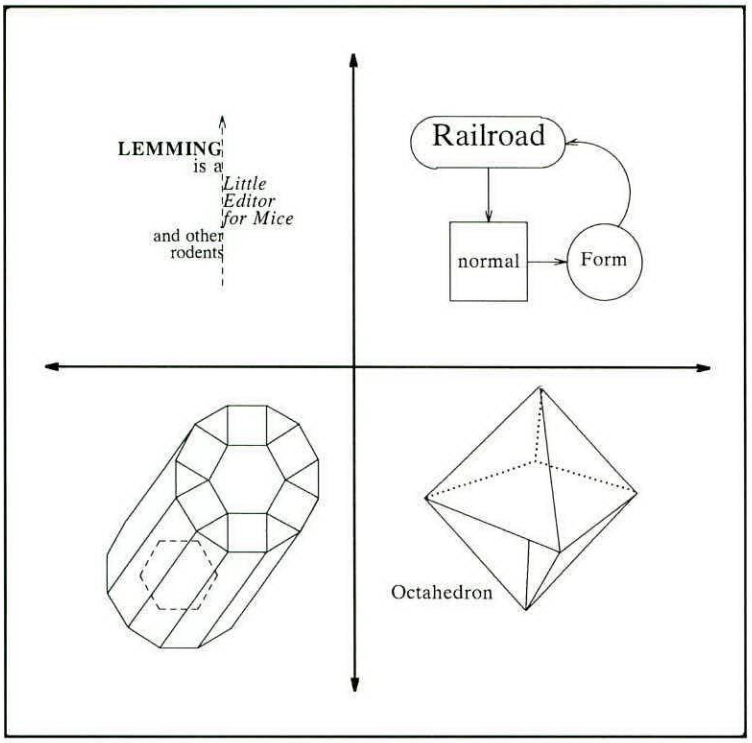


Paintings by De Chirico



Mondrian's *Broadway Boogie-Woogie*

The Center for Computer Animation is part of the art department at WPC. Both undergraduate and graduate programs in computer art and animation are offered. The center also serves to introduce students to real world situations in computer animation. A high level section of the laboratory works on actual productions for television.



lemming - little editor for mice and other rodents.

Alan W. Paeth
Computer Graphics Laboratory
Department of Computer Science
University of Waterloo, Waterloo
Ontario, Canada N2L 3G1
< AWPaeth%watCGL@Waterloo.CSNet>

The Lemming editor is an interactive package for modeless editing of simple (PIC-like) graphics images. Both PIC and Postscript output are supported. Designed to be highly portable, it minimally requires a one button mouse and binary (one bitplane) display to operate, yet benefits from systems supporting color, menus or graphics primitives. The IRIS implementation is particularly straightforward. Other systems (Adage/Ikonas, X-windows) make use of an internal library to convert vectors and arcs into raster primitives.

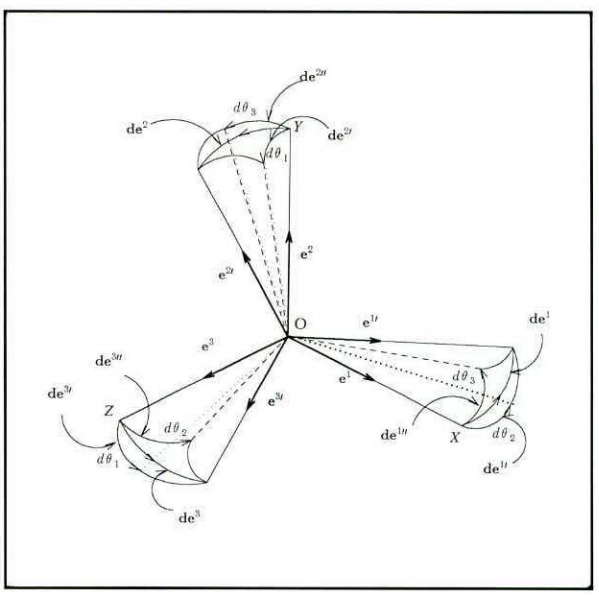
The objects supported include lines, text, circles, boxes with square or rounded corners and ellipses. Graphics attributes include dotted or dashed borders of specified thickness, text justification, font styles and sizes. The font package and object rendering code are completely self-contained to assure portability. The font library includes serif and sans-serif and special faces in a wide range of point sizes.

The editor includes a number of unique features. Both square and hexagonal gridding are supported at various pitches. Lines have the ability to scissor other lines, thereby facilitating the creation of hidden or highlighted edges. General transformations include rotation, mirroring, and scaling, plus a rectify operation used to axis align free-hand data. Facilities for object grouping, operation undoing and file manipulation make the editor more robust. The modeless interaction model allows for general object selection and manipulation with infrequent need to resort to the keyboard.

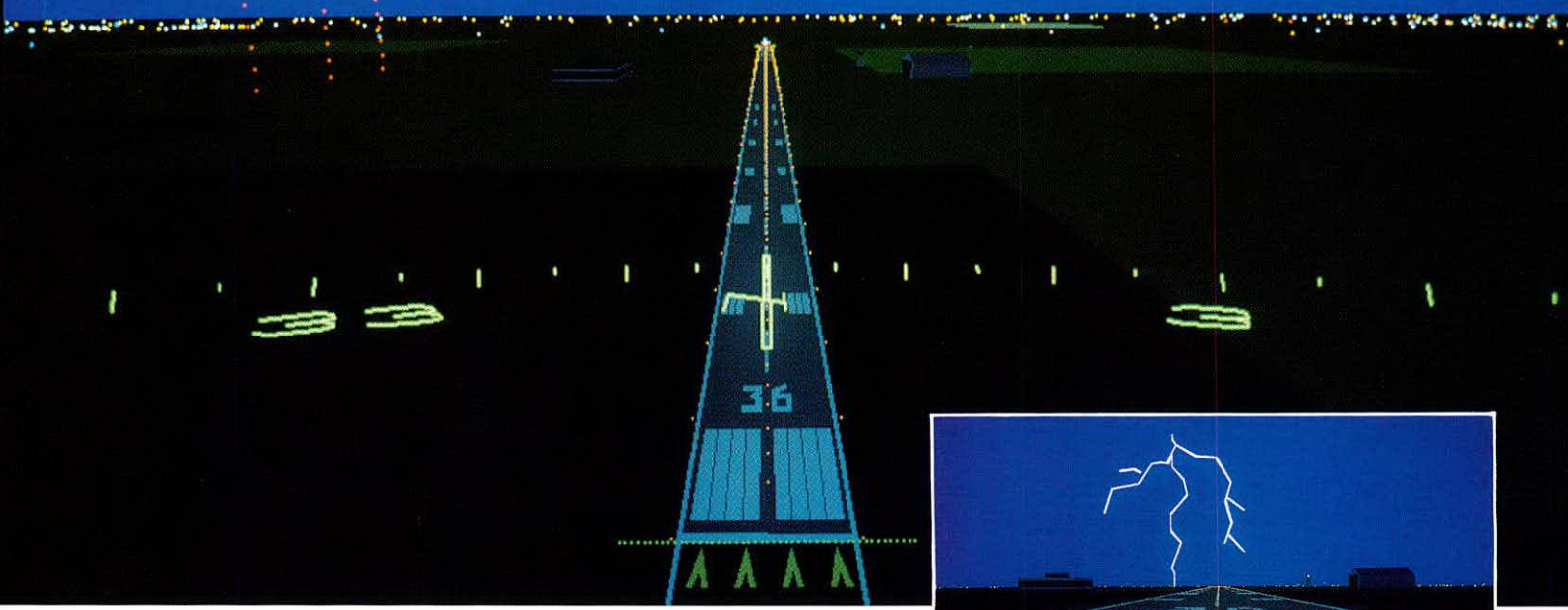
Lemming's pedigree is via "SIL", a line-only schematic editor created at Xerox PARC in the 70's. The present code is modeled after "Quicksilver" -- an award-winning desk-top publishing program designed and co-implemented by the author on the Alto personal computer. The latter was coded in Smalltalk-76; Lemming's C-code internals strongly reflect the object-oriented style of its ancestors.

Copies of the software and all documentation (sample documents, man pages, tutorial and installation guide) may be obtained via e-mail from the author.

incremental rotations



VISUAL STIMULATION



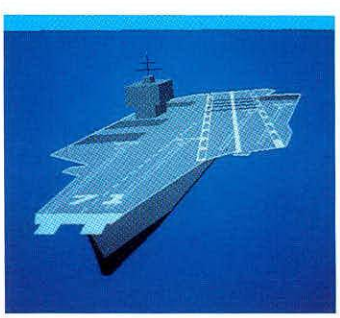
ONE SOFTWARE TOOL FOR ALL YOUR REAL-TIME VISUALIZATION NEEDS

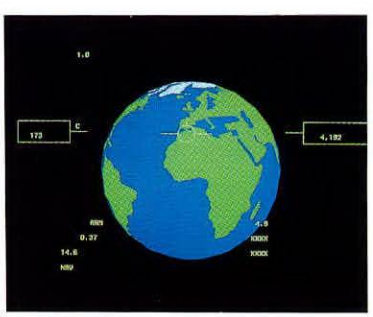
What do British Aerospace, McDonnell Douglas, TRW, Rockwell International, Northrop, the U.S. Navy, the U.S. Air Force, and other high-technology organizations have in common?

THE GENERIC VISUAL SYSTEM™

GVS software provides a powerful and cost-effective solution for embedded systems and computer generated imagery (CGI) applications. The GVS provides portable, real-time solutions for any graphics application, e.g., flight, ground, or space simulation, which requires fast display update rates.

- Create your own universe easily — add your own out-the-window scenes or 2-D and 3-D symbology applications
- Add your own special effects or use ours
- Supports your single or multiple channel applications
- Easily interface your external host computers — choose Ethernet TCP/IP, UDP, or shared memory network hardware available from Gemini Technology
- Interfaces with your *existing* simulation software
- Maintains graphics hardware operation at peak performance
- Automated level of detail scene management
- Automated airport scene generation tools
- 3-D object control is as simple as X-Y-Z, pitch-roll-heading
- Data bases—model directly with the GVS or import data base models from various off-line interactive modelling systems including MULTIGEN or SUPER 3D
- Hardware independence—PHIGS, PHIGS + compatible
- Rapid prototyping for real-time symbols including Head-Up, Multi-Purpose, or Helmet Mounted Displays
- Software development with Fortran-77, C, or Ada®



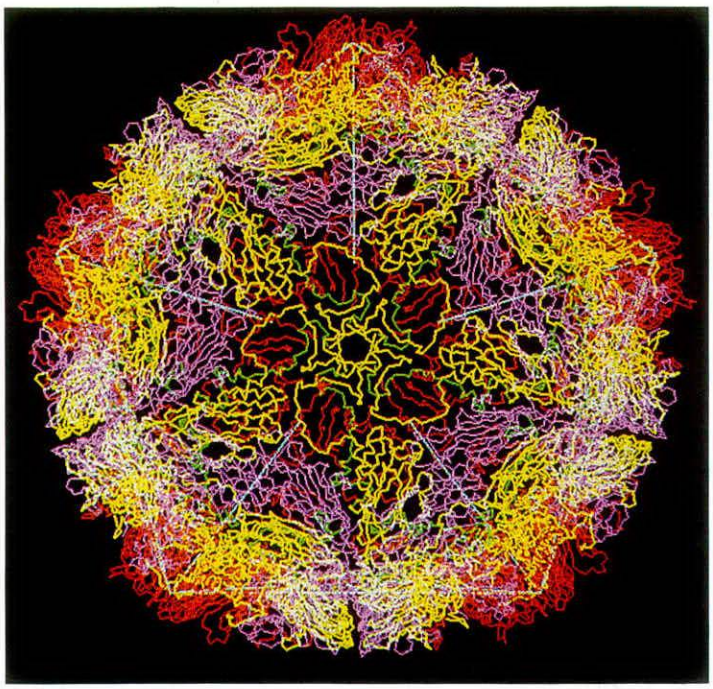


GVS

For information or a demonstration,

Gemini Technology Corp.
11770 Warner Avenue, #116
Fountain Valley, CA 92708 USA
Tel: (714) 966-8455
Fax: (714) 540-5261

Generic Visual System and GVS are trademarks of Gemini Technology Corporation. MultiGen is a trademark of Software Systems. Ada is a registered trademark of the Department of Defense. Super 3D is a trademark of Silicon Beach Software, Inc.



Rhinovirus (common cold virus) by T.J. O'Donnell. Data by Dr. Rossman, Crystallography Group, Purdue University

GRAMPS
a real-time, interactive, 3D picture editing and animation system

Maxine D. Brown
Electronic Visualization Laboratory
University of Illinois at Chicago

less is more: simplicity of design handles complex tasks

Molecular modeling animations *Tomato Bushy Stunt Virus, ADAM (A Dial-Activated Man), Terms of Entrapment* and *Poliovirus* are familiar works to SIGGRAPH Film and Video Show audiences. GRAMPS, a graphics software package used to create them, is well known to chemists and is used in a number of chemical applications, such as x-ray crystallography, reaction dynamics, statistical mechanics and quantum chemistry. And now that GRAMPS is being ported to IRIS machines, Silicon Graphics' users will have the opportunity to experience first-hand its power as a general-purpose real-time 3D picture editing and animation system.

GRAMPS co-author T.J. O'Donnell is converting the software to IRIS workstations (2000/3000/4D series) as part of a remote scientific visualization project being funded by the National Center for Supercomputing Applications (NCSA). O'Donnell, an independent consultant who studied computer graphics at the Electronic Visualization Laboratory (EVL) and received a Ph.D. in physical chemistry from the University of Illinois at Chicago (UIC) in 1980, has returned to EVL to conduct his research. He is cooperating with Scott Kahn of the chemistry department at the University of Illinois at Urbana-Champaign, and Dave Heisterberg of NCSA. Output from the molecular dynamics simulation package AMBER, running on NCSA's supercomputer, will be input to GRAMPS for display on remote IRIS workstations.

GRAMPS, a graphics language interpreter, was written by O'Donnell and Arthur Olson at the National Resource for Computation in Chemistry (NRCC) at Lawrence Berkeley Labs in 1980-81 on an E&S Multi-Picture System in FORTRAN '77. GRAMPS was developed to handle an increasing number and variety of chemical applications. O'Donnell recalls, "We wanted to develop a molecular modeling package that satisfied everyone's need. It had to be able to represent everything — from electrons and protons, to atoms and bonds, to macro-molecular 'schematic' representations — and anything else a chemist might want to display. To do this, we forgot about chemistry — we developed a system that knew nothing about chemistry, but was capable of handling the kinds of spatial and temporal animation needed for chemical applications. We provided a graphics tool for scientists."

hierarchical picture definition

By design, GRAMPS does not rely on molecular input. It accepts any 3D structure that can be expressed as *points, lines, polygons* and *spheres*. These primitives are used to build *objects*. One or more objects can be defined as a *group*. A group can be manipulated as a single entity, or its component objects can be manipulated individually. Objects and groups can be further grouped into a *hierarchical (nested) structure* and the resulting motion is *articulated* (the motion of any arbitrary group affects the motion of its subgroups). And, because GRAMPS is also an animation system, motion sequences can be played back, and the scene can be viewed from any angle in 3-space.

- DNA, for example, is comprised of a number of nucleotide bases. Using GRAMPS, a chemist first constructs a nucleotide base object. The chemist

then defines DNA as a hierarchical structure: grouping two nucleotide bases starts the formation of a polynucleotide chain; forming new groups, by adding another instance of the nucleotide base to the previous group, causes the chain to grow. The scientist can then manipulate the entire chain, parts of the chain, or individual objects within the chain. This type of interaction enables the chemist to rotate the various subchains around the molecule's glycosidic bond to simulate the effects of unwinding the DNA.

• To show how general purpose GRAMPS was, O'Donnell and Olson created ADAM, a 22-jointed human figure. ADAM's parts are defined in a hierarchical structure, and all body parts defined within a group follow the motion of the group. For example, the torso is a group of five body segments (analogous to a spine that can bend at five locations) plus the arms and head. The torso segments are grouped from the waist up, each segment being grouped with those above. This structure not only ensures the proper following of all the appropriate body parts when the torso is bent at the waist, but also (in a similar fashion to the DNA example) uses nested rotations to emulate complex spinal bending and twisting. Body motions and motion limits can be assigned to input dials on the workstation console, giving the user real-time control over the model's movements.

picture creation, manipulation and animation: GRAMPS feature set

Also by design, GRAMPS combines ease-of-use with powerful flexibility. It enables scientists with little or no graphics knowledge to interactively display, manipulate and animate 3D geometric picture files.

• *Creating 3D picture files:* Input to GRAMPS can be precomputed using simulation programs or can be hand-drawn using a tablet or mouse. "Images are usually computed," points out O'Donnell, "because scientists depend upon the mathematical accuracy of their simulations to understand what's happening."

• *Manipulating picture databases:* Users can define a variety of motions, as well as operations on those motions. A sequence of motions (a *script*) can be predefined and stored in a file for subsequent viewing, or movements can be defined interactively and saved. In the latter case, motions and motion limits for specific objects in a scene can be assigned to dials and knobs on the console, giving users dynamic control of the movements. "Scientists want to be able to explore," continues O'Donnell, "they want to figure out what they want to do while they are doing it."

Motion commands include *rotate, translate, scale, color* and *color intensity* (for depth cueing). For each motion, we can define *interpolation* (the number of time steps over which the motion is to be performed), *boundary conditions, speeds*, etc. GRAMPS also has a powerful macro feature that enables new graphics commands (such as complex motion sequences) to be developed and incorporated interactively. There is also a *perspective command* that defines the front and back clipping planes of the entire scene being displayed; by interactively modifying the clipping plane, one can essentially cut through the objects in the scene, exploring their volumes.

• *Animation of objects:* Objects can be manipulated on the workstation display screen in real time — a scientist either can use knobs and dials to manipulate the objects interactively, or can create and execute a command file. When defining movements dynamically, scientists can save *key frames* (a sequence of key positions of objects within a scene) to a file and then use that file to interpolate between those frames for final viewing.

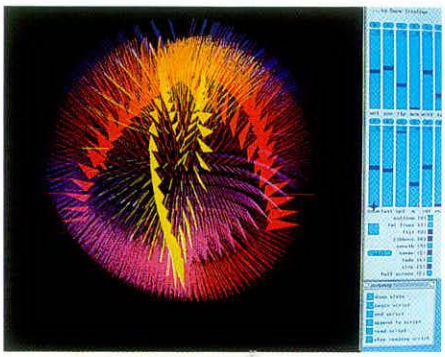
"Molecules, however, are not rigid structures. Atoms wobble around, configurations change when enzymes dock with substrates, etc. Chemists are more interested in how molecules move than how they look; in fact, no one knows what a molecule really looks like. The ability to visualize the effects of molecular dynamics, or vibrations, is critical," explains O'Donnell, "so we included a very powerful animation feature in GRAMPS that we called *framed objects*."

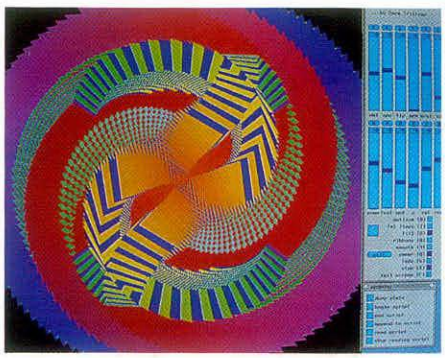
Simulation programs are capable of computing structures as a function of time. As a structure changes, its geometric description changes. A *framed object* is one whose description changes over time. For a given object, GRAMPS stores a sequence of descriptions along with time-dependent information. When an animation is played back, GRAMPS automatically inserts the appropriate geometric descriptions of framed objects at the appropriate points in time. This capability, used in conjunction with key frame animation, creates very complex — and very revealing — motion displays.

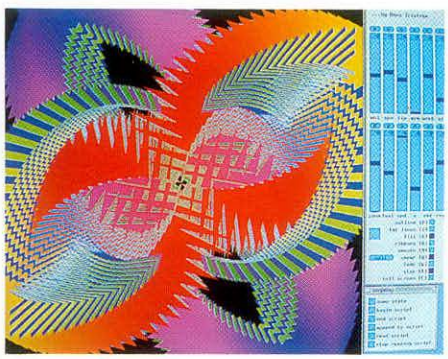
These animations can then be played in real time, and chemists can study whether or not they produce the desired effects. After a sequence is judged to be correct, it can be played back in real time or synchronized with an animation camera for filming.

documentation

The software will be available for distribution on IRIS machines in the Fall of 1988. In addition to porting the existing package to new workstations, O'Donnell is busy making enhancements and adding new features. For further information, write T.J. O'Donnell, Electronic Visualization Laboratory (M/C 154), University of Illinois at Chicago, Box 4348, Chicago, IL 60680.







electro-paint and panel library

David A. Tristam
NASA Ames Research Center
<dat@orville.nas.nasa.gov>

Electro-Paint is a colorful and dynamic interactive graphics application demonstrating the use of the Panel Library, a user-interface manager for the IRIS family. The user of Electro-Paint moves sliders and operates buttons with the mouse to control the evolution of a series of triangles on the screen. Incremental rotations, translations, and scaling are dynamically applied to an original leading layer and then are propagated through as many as 128 subsequent layers of triangles. Other mouse-sensitive actuators control drawing parameters such as outlines, full-screen mode, and smearing. The actuators provided by the Panel Library are completely independent of Electro-Paint and may be used in other programs.

Electro-Paint and the Panel Library were written by David A. Tristram of the NASA Ames Research Center. Electro-Paint includes code written by Paul Haeberli of Silicon Graphics. The software is in the public domain and may not be resold. Any use of Electro-Paint in public performance or demonstration must be accompanied by the acknowledgement above.

Both pieces are available on the Software Exchange Release Tape #1.
See page 41 in this isssue for further information about the IRIS Software Exchange, or contact:

Monica Schulze
Silicon Graphics, Inc.
2011 N. Shoreline Blvd.
Mountain View, CA 94039-7311
(415) 962-3320
user@sgi.com

QSH: a portable image display and processing system

Gerald Q. Maguire Jr.
Dept. of Computer Science
Columbia University

Marilyn E. Noz
Department of Radiology
New York University
550 First Avenue, New York, NY 10016
!cmcl2.nyu.edu!nucm1!noz

archiving and handling of images

In any environment in which different types of images are produced, there are many common tasks which are required to handle these images. Following image production either in analog or digital form, there may be a need to enhance or otherwise process the image, make it available for use (e.g., viewing), store it for later retrieval, and transmit it to different locations. In radiology, the conventional approaches to the performance of these tasks has led to workable but inefficient solutions. Most of these solutions revolve about the use of film. However, film retrieval for comparison or research use is often hampered by film misplacement and the high personnel cost of filing, retrieving and transporting the film; radiologists and clinicians often have to physically travel to the image source locations; space requirements often dictate that older films be stored off-site, and sending films to other institutions involves reproduction, shipping cost and delays.

then defines DNA as a hierarchical structure: grouping two nucleotide bases starts the formation of a polynucleotide chain; forming new groups, by adding another instance of the nucleotide base to the previous group, causes the chain to grow. The scientist can then manipulate the entire chain, parts of the chain, or individual objects within the chain. This type of interaction enables the chemist to rotate the various subchains around the molecule's glycosidic bond to simulate the effects of unwinding the DNA.

• To show how general purpose GRAMPS was, O'Donnell and Olson created ADAM, a 22-jointed human figure. ADAM's parts are defined in a hierarchical structure, and all body parts defined within a group follow the motion of the group. For example, the torso is a group of five body segments (analogous to a spine that can bend at five locations) plus the arms and head. The torso segments are grouped from the waist up, each segment being grouped with those above. This structure not only ensures the proper following of all the appropriate body parts when the torso is bent at the waist, but also (in a similar fashion to the DNA example) uses nested rotations to emulate complex spinal bending and twisting. Body motions and motion limits can be assigned to input dials on the workstation console, giving the user real-time control over the model's movements.

picture creation, manipulation and animation: GRAMPS feature set

Also by design, GRAMPS combines ease-of-use with powerful flexibility. It enables scientists with little or no graphics knowledge to interactively display, manipulate and animate 3D geometric picture files.

• *Creating 3D picture files:* Input to GRAMPS can be precomputed using simulation programs or can be hand-drawn using a tablet or mouse. "Images are usually computed," points out O'Donnell, "because scientists depend upon the mathematical accuracy of their simulations to understand what's happening."

• *Manipulating picture databases:* Users can define a variety of motions, as well as operations on those motions. A sequence of motions (a *script*) can be predefined and stored in a file for subsequent viewing, or movements can be defined interactively and saved. In the latter case, motions and motion limits for specific objects in a scene can be assigned to dials and knobs on the console, giving users dynamic control of the movements. "Scientists want to be able to explore," continues O'Donnell, "they want to figure out what they want to do while they are doing it."

Motion commands include *rotate, translate, scale, color* and *color intensity* (for depth cueing). For each motion, we can define *interpolation* (the number of time steps over which the motion is to be performed), *boundary conditions, speeds,* etc. GRAMPS also has a powerful macro feature that enables new graphics commands (such as complex motion sequences) to be developed and incorporated interactively. There is also a *perspective command* that defines the front and back clipping planes of the entire scene being displayed; by interactively modifying the clipping plane, one can essentially cut through the objects in the scene, exploring their volumes.

• *Animation of objects:* Objects can be manipulated on the workstation display screen in real time — a scientist either can use knobs and dials to manipulate the objects interactively, or can create and execute a command file. When defining movements dynamically, scientists can save *key frames* (a sequence of key positions of objects within a scene) to a file and then use that file to interpolate between those frames for final viewing.

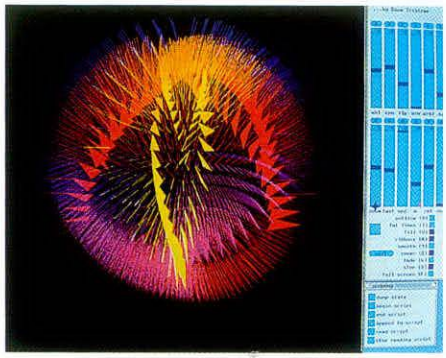
"Molecules, however, are not rigid structures. Atoms wobble around, configurations change when enzymes dock with substrates, etc. Chemists are more interested in how molecules move than how they look; in fact, no one knows what a molecule really looks like. The ability to visualize the effects of molecular dynamics, or vibrations, is critical," explains O'Donnell, "so we included a very powerful animation feature in GRAMPS that we called *framed objects*."

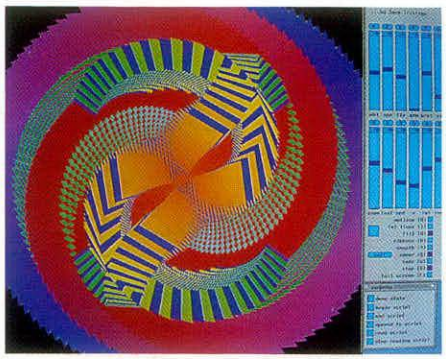
Simulation programs are capable of computing structures as a function of time. As a structure changes, its geometric description changes. A *framed object* is one whose description changes over time. For a given object, GRAMPS stores a sequence of descriptions along with time-dependent information. When an animation is played back, GRAMPS automatically inserts the appropriate geometric descriptions of framed objects at the appropriate points in time. This capability, used in conjunction with key frame animation, creates very complex — and very revealing — motion displays.

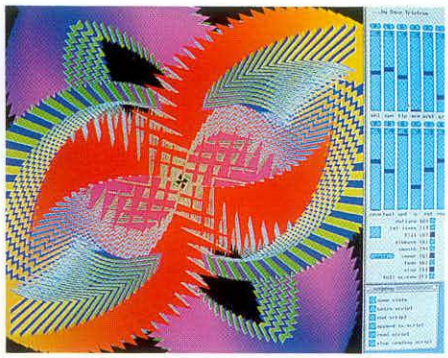
These animations can then be played in real time, and chemists can study whether or not they produce the desired effects. After a sequence is judged to be correct, it can be played back in real time or synchronized with an animation camera for filming.

documentation

The software will be available for distribution on IRIS machines in the Fall of 1988. In addition to porting the existing package to new workstations, O'Donnell is busy making enhancements and adding new features. For further information, write T.J. O'Donnell, Electronic Visualization Laboratory (M/C 154), University of Illinois at Chicago, Box 4348, Chicago, IL 60680.







electro-paint and panel library

David A. Tristam
NASA Ames Research Center
<dat@orville.nas.nasa.gov>

Electro-Paint is a colorful and dynamic interactive graphics application demonstrating the use of the Panel Library, a user-interface manager for the IRIS family. The user of Electro-Paint moves sliders and operates buttons with the mouse to control the evolution of a series of triangles on the screen. Incremental rotations, translations, and scaling are dynamically applied to an original leading layer and then are propagated through as many as 128 subsequent layers of triangles. Other mouse-sensitive actuators control drawing parameters such as outlines, full-screen mode, and smearing. The actuators provided by the Panel Library are completely independent of Electro-Paint and may be used in other programs.

Electro-Paint and the Panel Library were written by David A. Tristram of the NASA Ames Research Center. Electro-Paint includes code written by Paul Haeberli of Silicon Graphics. The software is in the public domain and may not be resold. Any use of Electro-Paint in public performance or demonstration must be accompanied by the acknowledgement above.

Both pieces are available on the Software Exchange Release Tape #1.
See page 41 in this isssue for further information about the IRIS Software Exchange, or contact:

Monica Schulze
Silicon Graphics, Inc.
2011 N. Shoreline Blvd.
Mountain View, CA 94039-7311
(415) 962-3320
user@sgi.com

QSH: a portable image display and processing system

Gerald Q. Maguire Jr.
Dept. of Computer Science
Columbia University

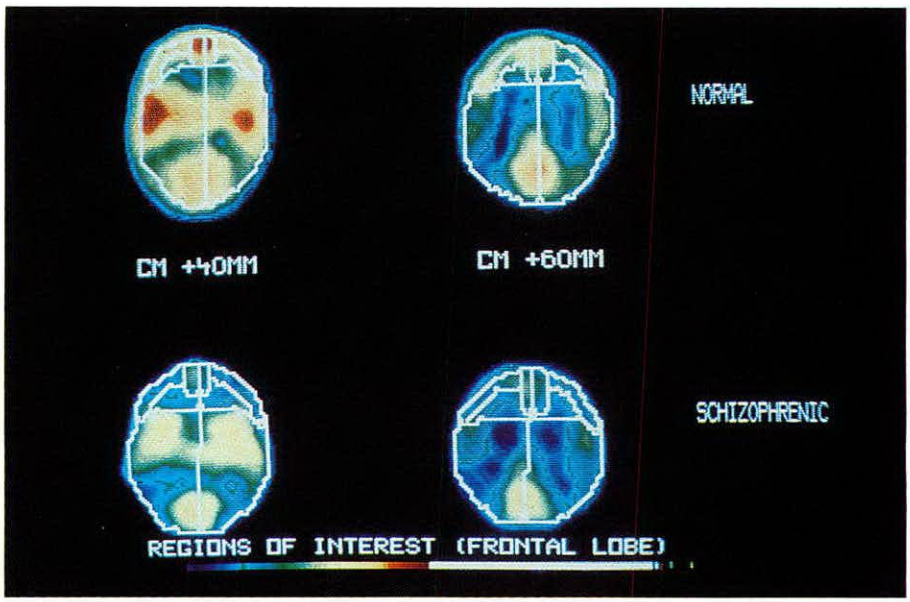
Marilyn E. Noz
Department of Radiology
New York University
550 First Avenue, New York, NY 10016
!cmcl2.nyu.edu!nucm1!noz

archiving and handling of images

In any environment in which different types of images are produced, there are many common tasks which are required to handle these images. Following image production either in analog or digital form, there may be a need to enhance or otherwise process the image, make it available for use (e.g., viewing), store it for later retrieval, and transmit it to different locations. In radiology, the conventional approaches to the performance of these tasks has led to workable but inefficient solutions. Most of these solutions revolve about the use of film. However, film retrieval for comparison or research use is often hampered by film misplacement and the high personnel cost of filing, retrieving and transporting the film; radiologists and clinicians often have to physically travel to the image source locations; space requirements often dictate that older films be stored off-site, and sending films to other institutions involves reproduction, shipping cost and delays.

digital imagery

In the case of imaging modalities which are inherently digital, such as computer assisted tomography (CT), positron emission tomography (PET), single photon emission tomography (SPECT), nuclear magnetic resonance imaging (NMRI) and digital radiography (DR) in its various forms, the situation is worse; there is replication of hardware costs in having separate viewing systems for each modality, there is loss of image information when these images are archived on film, the image formats are all incompatible and once the machine producing the image is retired as obsolete, those images obtained on it are often (nearly always) no longer available in digital form.



The figure shows two PET slices from a normal control and two from a schizophrenic subject with Regions of Interest (ROIs) drawn in the region of the frontal lobe. The glucose metabolic rate reduction in the schizophrenic subject is one characteristic of the disease. (The images are from the BNL/NYU PETT Project.)

common image format

The present authors started to grapple with these problems toward the end of the last decade. We realized that the problem existed on many levels. First, we needed a common image format to exchange images inter- and intramurally. We needed a common image format to display and manipulate images from different modalities on the same viewing console, i.e., so one set of application programs would apply to any image, as well as so that further application programs could be developed as new information was obtained from the images. Furthermore, we needed to support different display systems, we needed to obtain the data and we needed a method of taking advantage of hardware upgrades without having to re-write all the software.
Note that our solutions apply to any type of images and are not limited to the medical environment.
The authors along with many others proposed that the acquisition and storage problems be solved by implementing a network to interconnect imaging facilities and by encouraging broad cooperation which would lead to industry-wide standards. (Since that time the ACR-NEMA Digital Imaging and Communications Standard 300 has been published.) The image format described in this paper can be easily adapted to any standard format which is adopted in the future. We took advantage of the UNIX (UNIX is a Trademark of Bell Laboratories) operating system to give us hardware independence and chose to program in a high level language (in our case C) so as to easily go from one hardware environment to another. (Presently our routines support UNIX System V and UNIX BSD 4.x as well as EUNICE.)

the QSH system

Qsh is a set of subroutines and programs which uses a common format for image data. It may profitably be viewed as a host for the user's image applications programs. It accomplishes this by providing

- a common file format for images,
- a set of low-level image handling subroutines,
- and structural restrictions on the way programs may access images.

The format is defined as a file of type IMAGE. The standard format provides for the storage of auxiliary information, such as the history of the image. These standard images are then manipulated by a set of user-callable system procedures. The procedures reduce the need for programming the low-level input/output by providing these facilities. Hence, one of the principal uses for the *qsh* system is the provision of a structure for application programs written by the user. These programs operate upon images producing new images. It is required that all files operated upon by application programs be in the standard IMAGE format. The first implementation of *qsh* was done at the Karolinska Institute in Stockholm, Sweden.

software access

The complete paper, the QSH software, incl. man pages can be requested from Dr. Marilyn Noz. Please provide a cartridge for the software.

acknowledgements
This research was supported in part by an IBM Faculty Development Award and equipment grants from Hewlett-Packard and American Telephone & Telegraph Companies.

waterloo's dragon

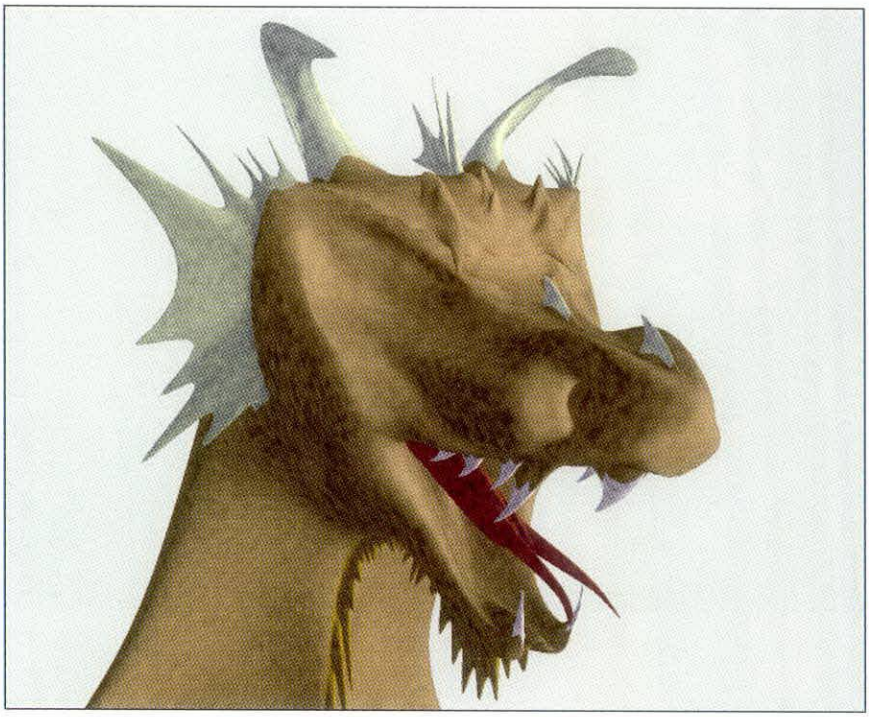
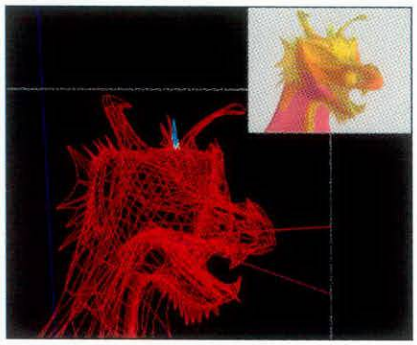
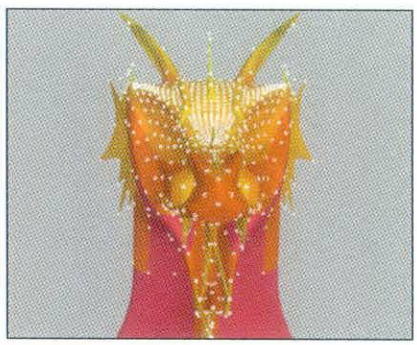


figure 1

These images were created on an IRIS 4D/70G utilizing a new technique for B-Spline manipulation (as explained in the paper "Hierarchical B-Spline Editing", Forsey & Bartels, SIGGRAPH 1988). An interactive editor was used to modify a torus into the dragon's head shown in Figure 1. The surface is defined by 6,505 control vertices. Using the traditional B-Spline representation would require over 36,000 control vertices.

To reduce confusion in the wire-frame presentation of the surface (Figure 2), the control graph and vertices are eliminated and replaced with a set of "edit points" that lie on the spline itself. The 6505 vertices are effectively represented by about 500 edit points (shown in Figure 3 as white "spheres"). Each edit point controls multiple control vertices. The different colours on the surface indicate areas with differing levels of detail - the lighter the colour, the finer the control over the surface within that area.





figures 2&3

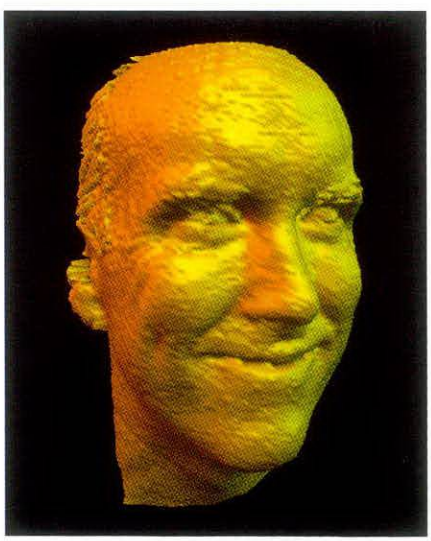
The shaded images required about 10 minutes to render. Stewart Kingdon supplied the texturing software.

All images are copyright by David Forsey and the Computer Graphics Laboratory, University of Waterloo.

talking head

Realistic character animation, in particular that of the human face, has always been one of the most difficult challenges in traditional or computer animation. It demands not only an extraordinary fidelity in the depiction of the character, but also careful imitation of the intricate interplay and motion of the expressive features of the human face (eyes, eyebrows, lips, cheeks, tongue, etc). Taking facial animation in the direction of higher-interactivity rather than increased realism, provided a challenging alternative to traditional frame-by-frame animation.

The original concept for the project arose out of a desire to create a complex animation piece that has a very strong interactive 3D graphic component. Silicon Graphics was in search of a project that could adequately demonstrate the dramatic new level of performance of its 4D/GT graphic system. The goal was to create an innovative demonstration that could be at once entertaining, technically challenging, and undeniably state-of-the-art.



Cyberware Laboratory's 3-D laser digitizing captured the complexities of a human face in various positions. Speech Systems's DS100 speech recognizition product creates a model for natural speech recognition, allowing phonetically labeled speech, Mike Gribble, computer animation promotor, volunteered to be cast as the head, and Michael Wahrmann and Brad DeGraf brought their extensive experience in character animation to bring Mike "to life", and connect him to interactive controls. The IRIS GT system provides the interactive performance to display a 3-D head which interactively 'talks'.
Watch for Mike-the Talking Head at SIGGRAPH.

You imagine something:

We picture it for you.

mental images

Gesellschaft für Computerfilm und Maschinenintelligenz mbH & Co. KG
Rankestrasse 9, D-1000 Berlin 30, Telephone 030/882 10 88, Fax 030/882 53 81,

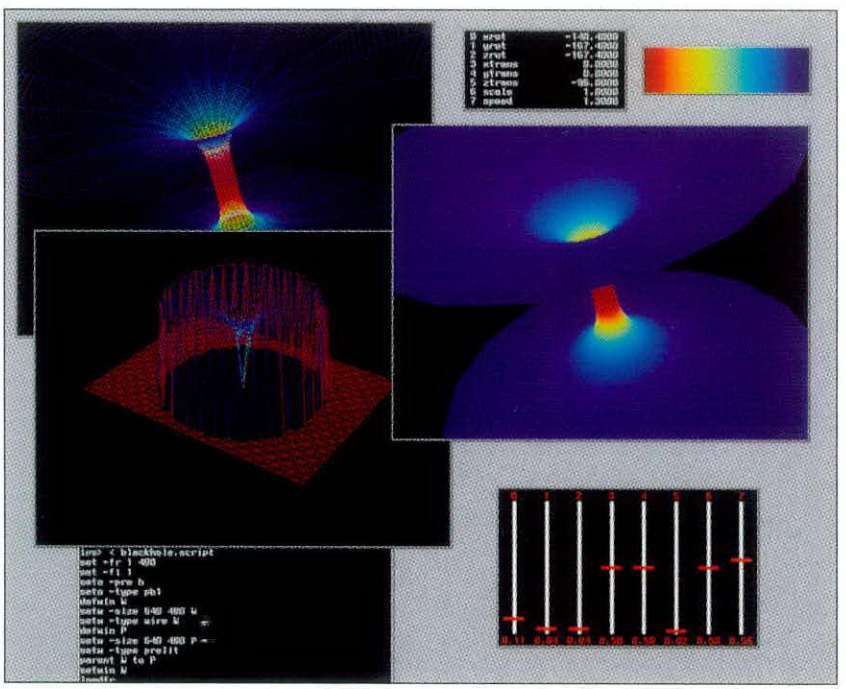
computer generated film and video, TV-trailers, music-clips, digital special effects, architectural design, industrial design - computer generated film and video for industrial and scientific purposes (process simulation, product descriptions, visual users manuals, visualisation of abstract or unobservable processes) - computer generated film and video for educational purposes - computer generated pictures for printing - software development for computer animation and for very fast image synthesis on parallel computers - software development for the simulation of physical processes, e.g. scattering and fluid dynamical processes.

Awards: PRIX ARS ELECTRONICA '87 (2.), BROADCAST MOTION GRAPHIC AWARD '87 (2.), NICOGRAPH '87 (3.), NCGA '88 (1.), PARIGRAPH '88 (1.)

curling up by the fire

This Christmas card was truly a team effort. The walls, floor and fireplace were modelled interactively by Steve Williams to create several hundred bicubic patches. The tree was created by Gavin Miller using a program which palced the branches and candles automatically. The problem was to devise an arrangement which prevented intersections between the candles and the branches. The fire was created using a procedural transparency map. Several surfaces were wrapped around the logs in a way which approximated the curving of the sheets of flame. These then had the transparency modulation applied to them. The snake skin was created using procedural texturing for the scales and painted imagery for the color markings.
Artists: Gavin Miller, Steve Williams, Pat Anderson, Gary Mundell, Andrew Pearce, Robert LeBlanc.
Alias Research Inc.

black hole at NCSA



Pictured on the left is one example of interactive scientific visualization being done at the National Center for Supercomputing Applications (NCSA) on the IRIS 4D superworkstation. The simulation represents a black hole (Prof. Larry Smarr principal investigator) where color denotes lapse time. In this particular example, the user can update parameters such as speed and orientation in real-time as the dynamics of the simulation are observed. These and other prototype computing environments are being investigated by the RIVERS (Research on Interactive Visual EnviRonmentS) project at NCSA. RIVERS is an interdisciplinary group of research scientists, faculty, staff, and students actively exploring ways to assemble advanced hardware, software, and networking systems for real-time visualization.

Silicon Graphics provides the hardware platform necessary for prototyping advanced computational visualization systems which integrate high-end graphics, giga-flop supercomputers, and massive high-speed networks. As development of these complex visualization systems continues, a framework of functionality will soon evolve allowing a researcher not only to view a simulation in real-time, but also to dynamically steer and process it while distributing portions of the simulation code to different machines which best accomodate it. (Illustrations provided by Ray Idaszak of the RIVERS project at NCSA, University of Illinois, Urbana-Champaign)

DOES YOUR PRODUCT NEED NURBS NOW

Eliminate 4-10 years of development time. Decrease **your** product's time to market. Increase your product's competitive position by providing better modeling and graphics. Come evaluate our NURBS (non-uniform, rational B-spline) library. Written in C, this library is designed to be easily integrated with applications in a wide spectrum of programming environments. Visit us in **Silicon Graphics' booth # 1236** during Siggraph. Or phone (801)575-6021.

Engineering Geometry Systems

275 E. South Temple, Suite 305
Salt Lake City, Utah 84111-1247
Telephone: (801) 575-6021

Foundations for Modeling Technology

a general rendering method

Paul Heckbert
CS Dept,
UC Berkeley
Berkeley CA 94708
<ph@miro.berkeley.edu>

Ray Idaszak's article "A Method for Shading 3-D Single-Sided Surfaces" in the winter/spring IRIS Universe suggests shading with a modified Lambert's law as a solution to certain tricky problems with consistent smooth shading of polygonal models. The article proposes shading with abs(N•L) instead of the traditional max(N•L,0), where N is the surface normal and L is the direction of the light from the surface point. Unfortunately, this shading formula effectively lights the object from opposite directions simultaneously, which would be unsatisfactory if one were trying to render scenes illuminated by sunlight, or pictures of a crescent moon, for example. Some alternative solutions to these shading problems are possible.

I will discuss faceted shading first, and then smooth shading.

The "semantics" or interpretation of a surface model is that of a shell: a solid model with infinitesimal thickness, shaded to be opaque. We must keep this in mind when writing rendering programs. We can honor these semantics as follows. If E is the direction of the eye from the surface point, then a back-facing point (or polygon) on a surface is one whose normal vector points away from the eye ($N \bullet E < 0$). For surface models of closed objects with a defined inside and outside ("orientable manifolds"), such as a sphere or torus, we can label the two sides of each polygon as interior and exterior, and the renderer can cull all back-facing polygons without artifact. This is possible since a polygon is back-facing if and only if we are viewing its interior side, and for a closed object we can't see the interior, by definition.

Culling back-facing polygons is invalid for non-closed (open) objects, such as uncapped cylinders, where the interior might be visible. For open objects, we must determine which side of the polygon to shade, since the surface is opaque and could have very different lighting on the two sides (outdoors on one side, indoors on the other, for example). We want to shade the visible side of the polygon, so we negate N if it is back-facing ($N \bullet E < 0$).

In contrast to Ray's method, we do this backface test once per polygon with respect to the eye direction E, not once per polygon per light with respect to the light direction L.

The above is fine for faceted shading, but smooth shading is trickier. How do we define the face normal if vertex normals are supplied, and how do we do back-face testing now that the normal is changing across a polygon? I encountered these issues while writing the z-buffer polygon renderer at the New York Institute of Technology (NYIT) several years ago, and I think I explored most of the possible dead ends before reaching a solution!

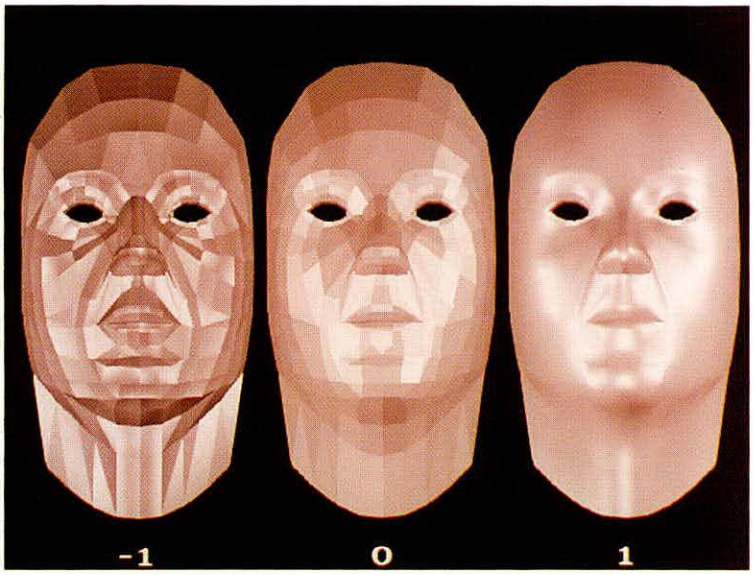
First consider: where are the vertex normals coming from? If they are coming from the model generator (a person or program) to specify them, it is reasonable to require that the model be consistent, since the generator presumably knows what surface they're trying to approximate. Consistency here means that the polygon vorticities should agree (all polygons clockwise, say, when viewed from the outside) in order that back-face testing work properly, and that the vertex normals agree with the vorticities (vertex normals point toward the clockwise side) so that shading is continuous. The other possible source of normals is an algorithm that infers normals from vertex coordinates. A common formula for the vertex normal is the average of the face normals around a vertex. If the generator has not supplied a polygon model with consistent vorticities then a vorticity propagation method such as Ray's is needed.

There are several possibilities for back-face testing of smooth shaded models:

(A) Take the face normal to be the average of the vertex normals. This doesn't work well, especially when the vertex normals are user-supplied and might not be well behaved. Say we're viewing the exterior side of a polygon near the silhouette of a sphere. Even if the user may have assigned it vertex normals so that the average vertex normal points slightly away from the eye, we should not shade the interior.

Moral: Take the face normal to be the normal to the plane containing the polygon, not the average of the vertex normals. The sign of the face normal will be determined by the vorticity of the polygon vertices. One can follow the convention, for example, that polygons go clockwise when viewed from outside the object.

(B) What about Phong shading? Do we determine front/back-facing on a pixel-by-pixel basis or once for each polygon? Pixel-by-pixel back-face testing is another disaster. For the same example above, with pixel-by-pixel testing we could get sparkling pixel artifacts near the silhouette of the sphere where the shading switches from exterior side to interior side.



Varying the vertex normal of a smooth shaded polygonal model. Vertex normals for smooth shading are usually the average of the face normals of the surrounding faces (right). If we allow a vertex to have a different normal for each polygon incident on it, we can interpolate the vertex normals toward the face normals and adjust the apparent curvature of each polygon. Interpolating to the face normal yields faceted shading (center) and extrapolating beyond yields concave-looking polygons (left) or bulging polygons (not shown). Multiple normals per vertex allows smooth shaded models with creases in controlled places, such as between the lips and around the nose in the picture at right. All pictures have the same geometry; only the normal vectors have been varied. Pictures rendered using the author's polygon renderer at NYIT, parameterized face models developed by Fred Parke. ©Paul Heckbert, NYIT

Moral: Back-face testing should be done once per polygon, not once per pixel.

(C) Face normal equals polygon plane normal, perform back-face test once per polygon. If the polygon is back-facing then negate all the vertex normals of (a copy of) that polygon before shading. This works fine, even for devious users who assign random vertex normals, and for bump mapping, where the normal vector is perturbed at each pixel by a texture map. This method has been used successfully for animation at NYIT for several years.

Note that with this smooth shading method it is important that the polygons all have the same vorticity when viewed from the outside and that vertex normals all point outward, or we might cull incorrectly or shade the wrong side. The above solves faceted and smooth shading for open or closed two-sided polygonal models.

Next we'd like to generalize the method to handle one-sided surfaces ("non-orientable manifolds") such as Mobius strips, which have no inside or outside. There is no consistent set of vorticities and vertex normals for a one-sided surface, however, so we'll need to change our polygon database and semantics.

To render a Mobius strip, for example, here's what I'd do: make an imaginary cut in the surface so that it is no longer one-sided, generate a model with consistent vorticity and normals for this modified surface, then glue the surface back together at the cut. The vertex normals from the two sides of the cut will disagree, so the simple polygon modeling database with one normal per vertex won't work. But if we change our polygon database to allow multiple normals per vertex, however, we can succeed.

There are two ways to do this:

(1) With each polygon store a bit vector with one bit for each vertex of the polygon, stating whether that vertex's normal should be negated when rendering this polygon, where 1 means negate and 0 means don't. For the Mobius strip above, these bits would all be 0 except on one side of the seam.

(2) A more general scheme allows a different vertex normal for each polygon incident on a vertex. This is a superset of method (1), and allows smooth-shaded models with sharp corners and creases as well.

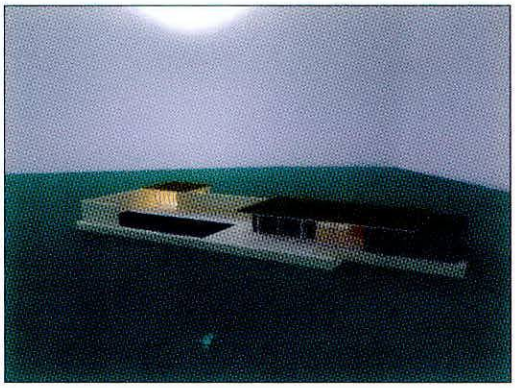
The latter method was used at NYIT. In the NYIT polygon modeling format, each polygon is described independently with a statement like:

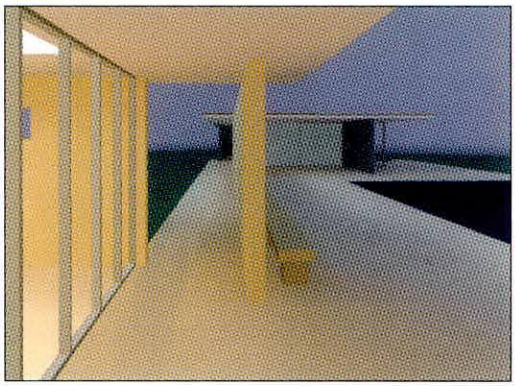
```
polygon (x1,y1,z1,nx1,ny1,nz1),
(x2,y2,z2,nx2,ny2,nz2),
(x3,y3,z3,nx3,ny3,nz3);
```

to describe a triangle, where (xi,yi,zi) is the vertex coordinate and (nxi,nyi,nzi) is the vertex normal.

While one must forego vorticity consistency for one-sided surfaces, it is possible (and essential) to maintain agreement between the polygon vorticity and vertex normals within each polygon in order for smooth shading to work correctly.

In conclusion, the above method will make smooth shaded renderings of one or two-sided polygon models while using the traditional shading formula.





The Barcelona Pavillion was designed in 1929 by Mies van der Rohe as the German pavillion for an International Exhibition. The pavillion was intended to be the exhibit rather than an edifice to house an exhibit, and is considered by many to be the most beautiful modern building to have been constructed anywhere. Few people, however, have experienced first hand Mies' subtle and elegant mastery of form and space as the pavillion was dismantled and lost in shipment on it's way back to Germany at the close of the Exhibition. It has been reconstructed recently, but limited photographs and the simplicity of it's 2D floor plan still make visualization of the Pavillion a frustrating experience for many who cannot journey to Barcelona. Real time, user controlled walk-throughs and real time lighting of the model, made possible by the performance of the Silicon Graphics 3D workstation, provide a much superior means of experiencing the Pavillion "first hand".
The radiosity team at SGI includes: Dan Baum, Efi Fogel, Dave Lignon, Jim Winget, Ben Garlick, Rolf Van Widenfeld.

real-time radiosity

These images demonstrate a photorealistic technique for rendering environments known as the Radiosity method. The Radiosity method is based on thermal engineering techniques for determining the exchange of radiant energy between surfaces. The application of these techniques to the field of computer graphics was pioneered by researchers at Cornell University.

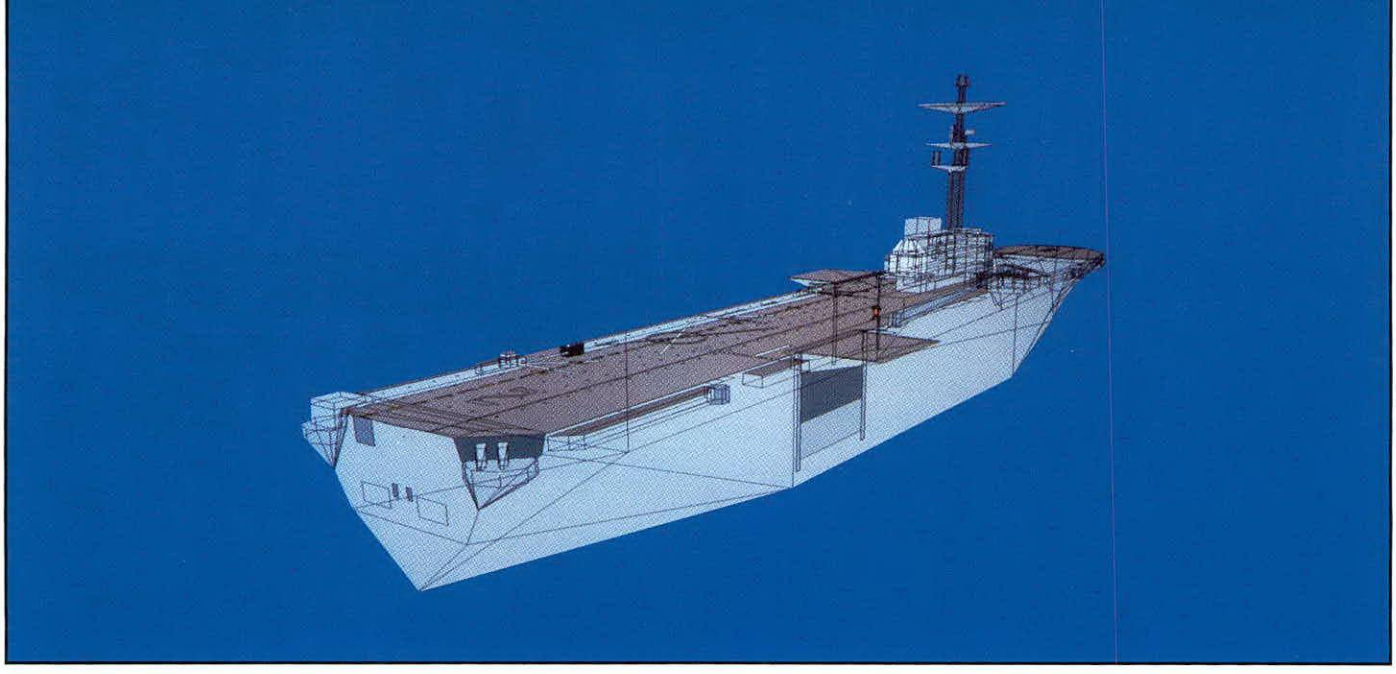
The method accurately computes the global illumination within a scene consisting of diffusely reflecting surfaces. Unlike local illumination models such as Phong, the Radiosity method takes the geometry and material properties of all surfaces in the scene into account when computing the illumination at any given point. Thus, effects such as shadows, indirect lighting, and color bleeding (reflected light assuming the color of the reflecting surface) are readily achieved.

The Radiosity method computes the illumination for environments consisting of diffusely reflecting surfaces. The result of this process is the illumination for the environment independent of view point. Once the initial illumination computation has been performed for an environment, that environment can be rendered from any point of view without performing any additional lighting calculations. Thus, after performing the initial illumination calculation, walk-throughs of an environment can be achieved in real time. This interactive walk-through ability makes the Radiosity method well suited for applications such as architecture, industrial design, and animation.

The Radiosity method computes an equilibrium energy balance for visible energy in the environment. To arrive at this equilibrium, the geometric relationships between every pair of surfaces in the environment must be computed. Once these geometric relationships have been found, they are combined with the reflectance values for each surface as well as the emmision values of light sources to form a large system of simultaneous equations. For an environment consisting of N surfaces an N by N matrix must be formed and solved.

In the past, people have refrained from using the Radiosity method to render images because of the complexity of the initial illumination computation. This computation may take several CPU hours on today's super minicomputers for moderately complex environments.

Recently, researchers at Silicon Graphics have developed a parallel Radiosity algorithm that greatly reduces the time to perform the Radiosity illumination calculation. The parallel algorithm exploits Silicon Graphics new multiprocessor superworkstation architecture as well as the capabilities of the superworkstation's proprietary 100 MFLOP graphics pipeline. Radiosity illumination computations that once took hours are now performed in a few minutes. Once the illumination computation has been completed, the user can walk through the environment using a Silicon Graphics superworkstation at rates up to 30 frames a second. Take a look at real-time radiosity at the Silicon Graphics booth at SIGGRAPH'88 in Atlanta.

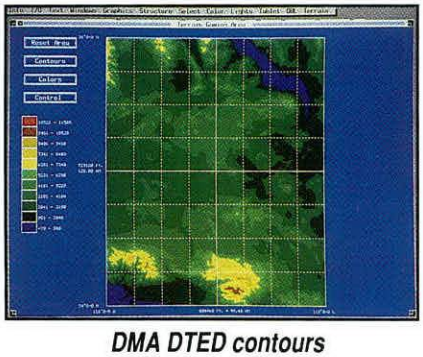


Would you like to create visual scenes in hours instead of weeks?

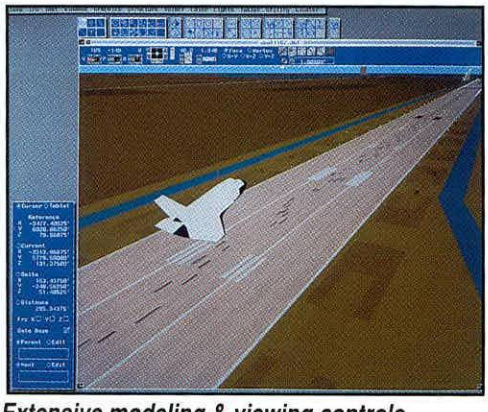
MultiGen ™ *gives you a fast, easy way to create and modify visual simulation data bases.*

MultiGen is a graphical editor for creation, modification and viewing of visual simulation scenes. Rapid, precise, WYSIWYG style data base modeling is made possible by an intuitive graphic interface and a rich set of 3D modeling tools designed specifically for visual simulation.

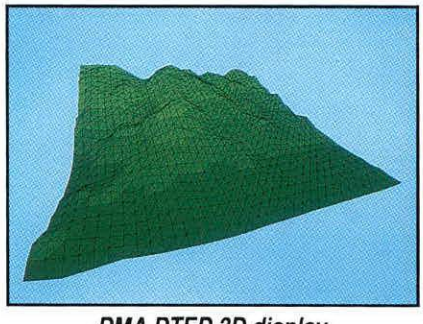
Software Systems developed MultiGen to address the requirements of both high-end image generators, as well as cost effective, IRIS-based real time systems. An elegant, public domain data base format (called Flight) is now available that is ideally suited for low to medium fidelity visual systems. Further, since data bases developed with MultiGen can be stored in a standard format, it is now possible to transfer models between different visual systems supported by MultiGen's Import/Export option.



DMA DTED contours



Extensive modeling & viewing controls



DMA DTED 3D display

The DMA DTED conversion option automatically generates 3D polygonal terrain from DMA elevation data. This terrain may then be populated with various objects and examined with MultiGen's assortment of viewing controls (including a fly-through capability), or "flown" with a compatible real-time system.

MultiGen integrates a set of powerful tools, options and graphical interfaces to make visual scene modeling fast and easy. It is available on the complete line of Silicon Graphics workstations.

Call to obtain sample data bases and a specification of the Flight visual data base format.

Software Systems *1625 The Alameda, Suite 700, San Jose, California, 95126 • Telephone 408-995-0689 • Facsimile 408-995-5058*

supporting your needs

Anita Schiller, Silicon Graphics Inc.

Undisturbed by the frantic pace of cars and trucks racing by on Highway 101 is the Silicon Graphics Product Support Group, known to many as the Geometry HOTLINE. You may already have had occasion to experience this group in action as someone hustled to help you solve a tough technical problem or answer a question about your IRIS Superworkstation. Even still, you may be curious to know who these people are and what they really do. So, read on...

The Product Support team is a group of highly skilled professionals dedicated to helping you achieve the height of success in the use of your Silicon Graphics Superworkstation. Expertise ranging from hardware to software and operating systems, with specialization in graphics, communications, and languages helps us to achieve our number one goal, customer satisfaction.

what's in it for you?

"What does Product Support do for me?", you might ask. In addition to providing direct telephone support to customers and Silicon Graphics' own field engineers, we are involved in a lot of behind the scenes activity that benefits current and future IRIS users. We help to isolate, verify and report bugs in Silicon Graphics products. We provide workarounds and fixes. Always with the customer in mind, we test new products and provide the product design and manufacturing teams with input on usability, reliability and supportability. In short, we want to ensure that the IRIS Superworkstation you buy is the best product it possibly can be.

directing the traffic

Approximately 350 new calls are opened each week. While the automobile traffic outside may be of little concern, the telephone traffic inside is of utmost importance. Getting the information for each call and directing it to the appropriate Product Support Engineer (PSE) is a full time activity for the team of Call Coordinators. They work hard to maintain order in what might otherwise be a chaotic scene. If you've called before, you know that one of the very first things you are asked is to provide your system serial number. This one small piece of information is the key to several on-line databases that may contain information about your configuration, support status, previous problems, etc.

the helpful team of the Geometry Hotline at SGI

keeping up with what's new

Committed to staying at the forefront of technology, the software PSEs frequently participate in programming projects with other groups in the company. We strongly believe that one of the best ways to increase the level of technical expertise within the group is through the use of Silicon Graphics' own products.

Speaking of programming projects, we know that many of you have been asking about the User Gifts Library being made available on the IRIS 4D Series Superworkstations. Well, this is the group working on expanding and porting the User Gifts Library to the IRIS 4D for you.

an eye on quality

Everyone knows that hardware engineers are interested in "what makes up the box". But did you know that with the PSEs, this may also mean the box in which your IRIS was shipped? The feedback we gather from customers on product shipments is used to improve the packaging and handling of systems to ensure that your IRIS Superworkstation is up and working sooner. In the more traditional sense, of course, we also track defects and work closely with Quality Assurance to develop corrective action plans. Our goal is to maximize the amount of time your system is available to you.

partners in success

You can help us to serve you better by being prepared whenever you call. In addition to knowing your serial number, record any error messages you may have received. Be prepared to describe the situation or environment in which the problem occurred (e.g. applications you were running, changes in hardware or software, examples,

any information that allows the problem to be duplicated, etc.). And, PLEASE, always check the documentation first if there is any question about what may have caused a problem.

The first available PSE with expertise related to your problem or question will return your call. It may be possible to resolve the problem or answer the question immediately, or it may be necessary to do some background research and call you back at a later time. In either case, you can be assured that Silicon Graphics has your success in mind. After all, your success is our success.

Silicon Graphics is striving to provide support that is legendary through a full complement of services such as the HOTLINE. But please keep in mind that the HOTLINE is not for everyone. It is available to those customers whose systems are covered by warranty or who have purchased maintenance contracts. For others, support is available on a time and materials fee basis. For more information on any of Silicon Graphics' Customer Support Programs, please contact your local sales representative.

Silicon Graphics Customer Support Center
Geometry HOTLINE - 800 345-0222
6:00 am to 5:00 pm - Pacific Time

IRIS software exchange

The IRIS Software Exchange is designed to make user developed software accessible within the IRIS User Community on a non-commercial basis. The entire spectrum of applications, including UNIX System V® programs, special graphics algorithms, file conversion programs, editors, drivers, languages and experimental software is covered in the Software Exchange. Silicon Graphics will continue to make selected demo programs available as part of the Software Exchange. Silicon Graphics also provides assistance in distributing suitable software across a variety of channels.

We are suggesting the following guidelines and welcome any feedback/input you may have. We encourage you to participate in the Software Exchange to increase the productivity of everyone in the IRIS User Community.

a) Software submitted to the Software Exchange is generally copyrighted by the author and not in the public domain; software will be accepted only from the author, or from another party with the written permission of the author.

b) No warranty of any kind is made in regard to the software, nor is any support provided for the software.

c) The authors and SGI will cooperate to ensure that each program submitted contains the following:

* source code
* makefile
* man page
* one paragraph description
* configuration specs

To order the first IRIS Software Exchange Tape, send check or purchase order for $100. Include your serial number and the name of IRIS series you are using.

Monica Schulze
Silicon Graphics Inc.
2011 N. Shoreline Blvd.
Mountain View, CA 94039-7311
(415) 962-3320
<user@sgi.com>

d) SGI will publish and update a catalogue of software available in the Exchange.

e) Several channels for distribution will be available:
* the authors distribute directly (NASA, BRL,etc)
* software in info-IRIS-Archives via FTP over the net from BRL-VGR.ARPA (192.5.23.6)
* cartridge distribution by SGI at nominal fee ($100)
* SGI established server accessable via modem

f) A representative will be appointed by IRIS users to help organize the Exchange.

Again, your feedback and help on this proposal are crucial. We are also interested in hearing from you about any software that you feel may be appropriate for possible inclusion in the IRIS Software Exchange.

Contributors to the IRIS Software Exchange will automatically receive copies of the release.

1st IRIS software exchange release tape is out

contents for IRIS 3000 series:

SGI demo source
Princeton toolkit
ICARE color editor
slidemaker
Naval Postgraduate School software

contents for IRIS 4D series:

SGI demo source
electro-paint & panel library
OPER remote job control
slidemaker

solutions...

Dorothy Liu, Silicon Graphics Inc.

Solutions. That's what the superworkstation environment provides you. But a solution isn't complete until everyone can partake of the benefits. That's why Silicon Graphics offers one of the most comprehensive training programs in the industry. By offering education to all our users, we make sure you don't miss any of the incredible capabilities the Silicon Graphics workstation possesses. If you don't have a total solution, you don't realize your total potential.

graphics gurus

Let your imagination soar. Designed for experienced applications programmers, our Advanced Graphics Programming course will help you achieve guru status. You'll learn some of the most sophisticated subroutines available and you'll model and render the fastest realistic images around. Your imagination comes to life when you learn advanced techniques such as anti-aliasing, alpha-blending and lighting models on the IRIS 4D/70GT.

system administration experts

Increase the power of your IRIS Superworkstation environment geometrically. Our Network Administration course, created for experienced system administrators, teaches you to incorporate industry networking tools into your computing environment. Taking advantage of the SGI workstation's interactive capabilities in the computing environment isn't an option; it's a requirement and we're prepared to help you meet that need.

new users

Share in the knowledge that many computer users already have. Our System Accelerator will teach new users basic operating system tools and commands and how to do first level system administration. You'll be up on our system faster and be able to utilize the tools you need to use your applications more effectively. You'll find that our superworkstation isn't just for graphics gurus and system administration experts but for the computer novice, too.

the program and facilities

With two fully operational Education Centers located in the San Francisco Bay Area and the Washington, DC area, as well as customer site training capabilities, Silicon Graphics provides training to both national and regional organizations. We can also customize our courses for you if you have special needs.

Solutions. That's what we're all about. To make your computing environment faster, more real and more complete.

The current schedule of classes is listed in the sidebar. For further information or to register for a class call the Customer Education Information Line: (800) 356-9492.

SILICON GRAPHICS INC. EDUCATION CENTER COURSE CALENDAR

COURSE	LOCATION*	
3000 SERIES	WEC	EEC
GRAPHICS I	Sept. 26, 1988 Nov. 28, 1988	not available
SYSTEM ADMIN	Aug. 29, 1988** Nov. 14, 1988	not available
SYSTEM MAINT (H/W Maint)	Sept. 12, 1988 Dec. 5, 1988	not available
4D SERIES COURSES		
GRAPHICS I	Aug. 15, 1988 Aug. 29, 1988 Oct. 24, 1988 Dec. 12, 1988	July 11, 1988 Aug. 15, 1988 Sept. 26, 1988 Oct. 10, 1988 Nov. 14, 1988
ADVANCED GRAPHICS **(new course)**	July 18, 1988 Aug. 22, 1988 Nov. 14, 1988	Sept.5, 1988 Oct. 17, 1988 Nov. 28, 1988
SYSTEM ACCELERATOR **(new course)**	July 11, 1988 July 25, 1988 Sept. 12, 1988 Oct. 3, 1988 Oct. 31, 1988	July 18, 1988 Aug. 8, 1988 Oct. 24, 1988 Dec. 12, 1988
SYSTEM ADMIN **(new after July 1)**	Aug. 1, 1988 Nov. 7, 1988	Aug. 22, 1988 Sept. 12, 1988 Dec. 5, 1988
NETWORK ADMIN **(new course)**	Aug. 8, 1988 Dec. 5, 1988	Aug. 29, 1988 Sept. 19, 1988 Oct. 3, 1988
SYSTEM MAINT	Sept. 19, 1988 Oct. 10, 1988	July 25, 1988 Oct. 31, 1988

KEY:
WEC--WESTERN EDUCATION CENTER,
SGI HEADQUARTERS, MOUNTAIN VIEW, CA.

EEC--EASTERN EDUCATION CENTER,
SGI FEDERAL, BETHESDA, MD

*The SGI Education Center reserves the right to cancel classes due to insufficient enrollment.

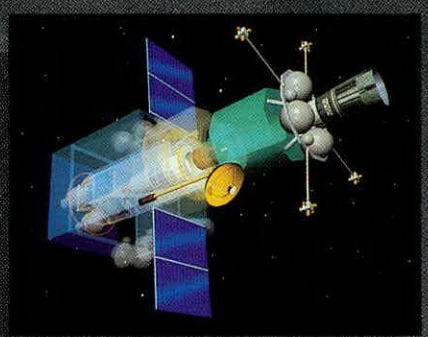
**Please note the change in date for the IRIS 3000 Graphics Programming Course

With Clients Like These, Who Needs Copy?

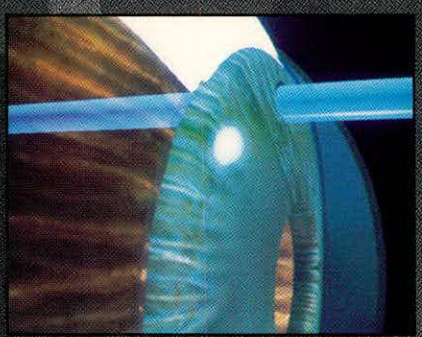
Partial
Engineering/
Scientific
Client List:

ADVANCED IMAGING
ATR COMMUNICATIONS SYSTEMS
AVERETT & ASSOCIATES
BARCO CREATIVE SYSTEMS
BIOGRAPHICS
BOEING ENGINEERING SCIENTIFIC
BOEING TECHNOLOGY CENTER
CHIYODA INSTITUTE OF TECHNOLOGY
COMPUTERVISION
CORNELL UNIVERSITY
ESTEC
FAILURE ANALYSIS ASSOCIATION
FORENSIC TECHNOLOGIES INC.
GENERAL DYNAMICS
HEWLETT PACKARD
HITECH LABS
INTEL CORPORATION
JET PROPULSION LABORATORY
MCDONNELL DOUGLAS
MORTON THIOKOL
NASA/DRYDEN
NASA/GODDARD
NASA/LBJ SPACE CENTER
NORTHROP RESEARCH AND TECHNOLOGY
NORTHROP AIRCRAFT DIVISION
NAVAL SURFACE WARFARE CENTER
NIPPON TELEGRAPH AND TELEPHONE
PRATT & WHITNEY
RTL PRODUCTIONS
SALT RIVER PROJECT
S.E.A.
SESA
SUMISHO ELECTRONIC SYSTEMS
TEXAS A&M UNIVERSITY
THE MITRE CORPORATION
TRW
UNIVERSITE LIBRE DE BRUXELLES
U.C.L.A.
UNIVERSITY OF ILLINOIS
UNIVERSITY OF MINNESOTA
UNIVERSITY OF TOKYO
U.S. AIR FORCE
VOLVO CAR CORPORATION

McDONNELL DOUGLAS MD11: DOUGLAS AIRCRAFT CORP. © 1988

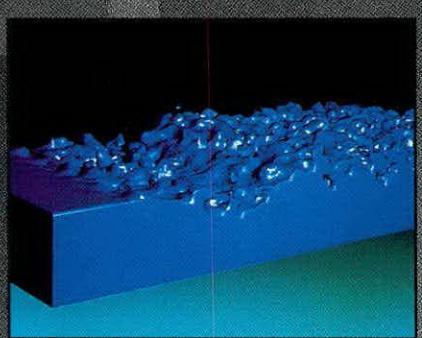


NASA/JSC/MPAD: "MARS SAMPLE RETURN MISSION"
INTERACTIVE GRAPHICS LAB DIRECTOR: GUNTER SABIONSKI
CHIEF ANIMATOR: ED KRAMER



MEDICAL RESEARCH: DAVID BOLINSKY
ADVANCED IMAGING © 1988

ROVER STERLING AUTOMOBILE
COMPUTERVISION: ICE'S DATABASE



"3D TURBULENT SHEAR FLOW AT MACH 1.1"
SCIENTIFIC RESEARCH: MICHAEL NORMAN, NCSA
VISUALIZATION: CRAIG UPSON, NCSA

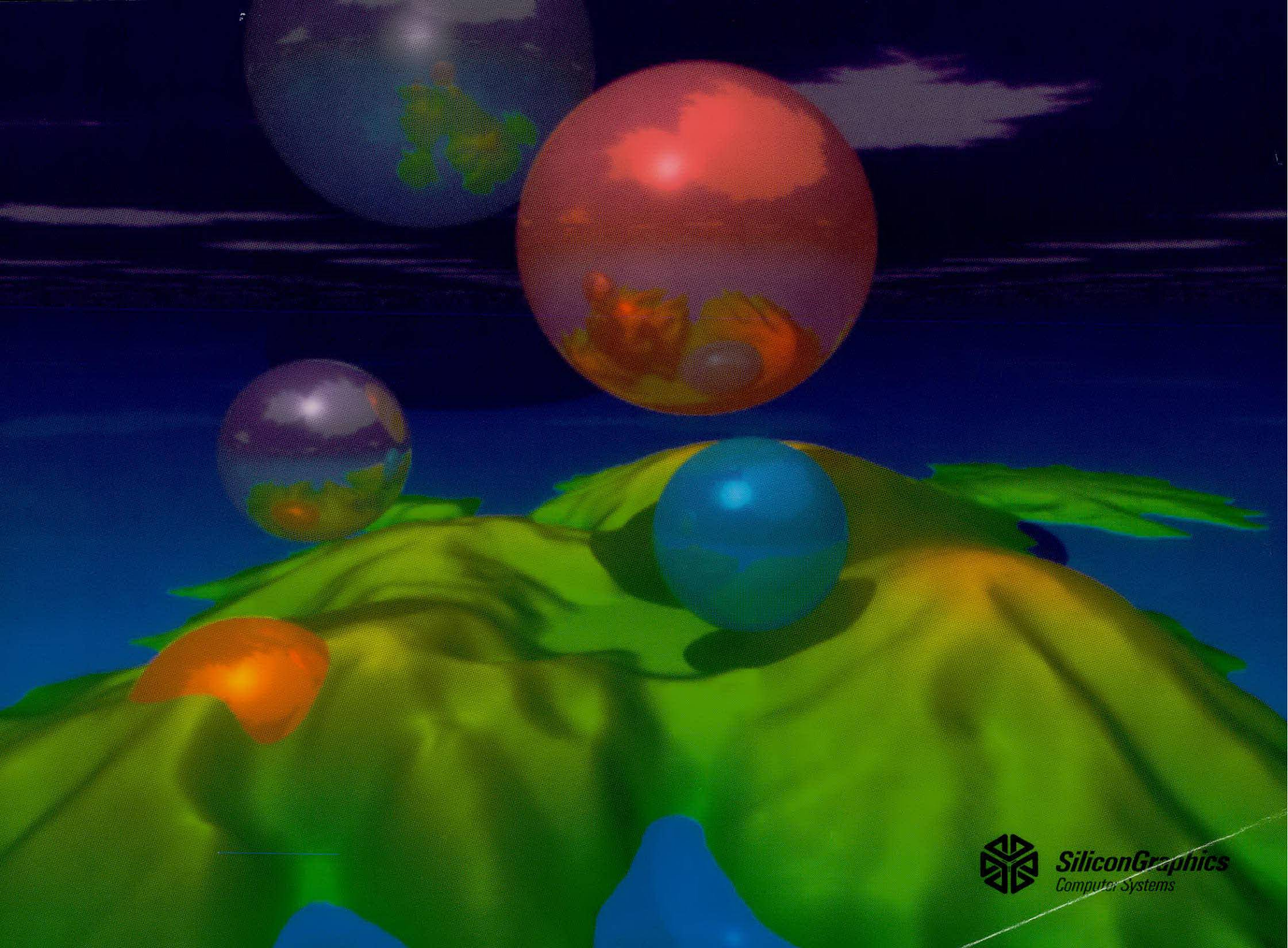
3D Software For Animation And Simulation.

Images Developed Utilizing Silicon Graphics Super Workstation.

WAVEFRONT TECHNOLOGIES

530 EAST MONTECITO STREET, SANTA BARBARA, CALIFORNIA 93103 805/962-8117

See What You Think.


SiliconGraphics
Computer Systems